Research Report 1403

Concept Study of the Computer-Aided ARTEP Production System (CAPS)

G. W. Bloedorn, W. H. Crooks, M. D. Merrill, H. J. Saal
Perceptronics, Inc.

L. L. Meliza and O. I. Kahn
ARI Field Unit at Presidio of Monterey, California

ARI Field Unit at Presidio of Monterey, California
Training Research Laboratory

ari

U. S. Army
Research Institute for the Behavioral and Social Sciences

July 1985

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Report 1403	2. GOVT ACCESSION NO. AD-A170951	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CONCEPT STUDY OF THE COMPUTER-AIDED ARTEP PRODUCTION SYSTEM (CAPS)		5. TYPE OF REPORT & PERIOD COVERED Final Report September 1984-May 1985 6. PERFORMING ORG. REPORT NUMBER --
7. AUTHOR(s) Bloedorn, G. W., Crooks, W. H., Merrill, M. D., & Saal, H. J. (Perceptronics); & Meliza, L. L., & Kahn, O. I. (ARI)		8. CONTRACT OR GRANT NUMBER(s) DABT60-84-C-0109
9. PERFORMING ORGANIZATION NAME AND ADDRESS Perceptronics, Inc. 6271 Variel Avenue Woodland Hills, CA 91367		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q263743A794 4311 100
11. CONTROLLING OFFICE NAME AND ADDRESS ARI Field Unit P.O. Box 5787 Presidio of Monterey, CA 93944-5011		12. REPORT DATE July 1985 13. NUMBER OF PAGES 175
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue Alexandria, VA 22333-5600		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE --

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

--

18. SUPPLEMENTARY NOTES

1. Contracting Officer's Representative, Otto I. Kahn.
2. Appendixes published separately.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Army Training and Evaluation Program (ARTEP)
Computer-aided ARTEP Production System (CAPS)
Relational data base management system

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Army service schools use typewriters and/or stand-alone word processors to prepare Army Training and Evaluation Program (ARTEP) documents as guides to unit training. Requirements to periodically revise ARTEP documents to reflect the continual changes in tactical doctrine associated with the force modernization effort, combined with the new requirement to prepare "improved" ARTEP documents containing detailed unit training plans, have greatly increased the size and complexity of the ARTEP development workload.

(Continued)

ARI Research Report 1403

20. (Continued)

The purpose of this research was to define the concept of a Computer-aided ARTEP Production System (CAPS) for the development of a prototype system within the U.S. Army Infantry School (USAIS). The functions to be assisted by automation were defined through reviews of TRADOC guidance for preparing "improved" ARTEPS, TRADOC Reg 310-2 (Test), and through review of the ARTEP development process within USAIS. Computer technologies appropriate to these functions were selected from among those technologies frequently applied to document preparation.

This report provides input for the selection of hardware/software for a prototype CAPS, and it aids in defining the research/development tasks required to develop such a prototype.

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON **Technical Director**	**WM. DARRYL HENDERSON** **COL, IN** **Commanding**

Research accomplished under contract for
the Department of the Army

Perceptronics, Inc.

Technical review by

Gean Bigler
Jean Dyer

A-1

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POT, 5001 Eisenhower Ave., Alexandria, Virginia 22333-5600.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Research Report 1403

Concept Study of the Computer-Aided ARTEP Production System (CAPS)

G. W. Bloedorn, W. H. Crooks, M. D. Merrill, H. J. Saal
Perceptronics, Inc.

L. L. Meliza and O. I. Kahn
ARI Field Unit at Presidio of Monterey, California

ARI Field Unit at Presidio of Monterey, California
Jack H. Hiller, Chief

Training Research Laboratory
Robert J. Seidel, Acting Director

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

July 1985

Army Project Number
2Q263743A794

Education and Training

ARI Research Reports and Technical Reports are intended for sponsors of R&D tasks and for other research and military agencies. Any findings ready for implementation at the time of publication are presented in the last part of the Brief. Upon completion of a major phase of the task, formal recommendations for official action normally are conveyed to appropriate military agencies by briefing or Disposition Form.

FOREWORD

The Army Research Institute (ARI) Presidio of Monterey Field Unit is concerned with improving unit collective training through research and development. One aspect of this work concerns the design and preparation of Army Training and Evaluation Program (ARTEP) documents by Army service schools as guides to unit collective training. Research in this area is conducted by the Collective Training Design Team under the sponsorship of the proponent for ARTEP development policy and procedures, the U.S. Army Training Board (ATB).

This report defines the concept of a Computer-aided ARTEP Production System (CAPS) in preparation for the development of a prototype CAPS within the U.S. Army Infantry School (USAIS). Research partners included in this effort include the U.S. Army Armor School (USAARMS), the U.S. Army Air Defense Artillery School (USAADASCH), and the U.S. Army Intelligence Center and School (USAICS).

EDGAR M. JOHNSON
Technical Director

CONCEPT STUDY OF THE COMPUTER-AIDED ARTEP PRODUCTION SYSTEM (CAPS)

EXECUTIVE SUMMARY

Requirement:

US Army service schools prepare Army Training and Evaluation Program (ARTEP) documents as guides to unit collective training. School workloads are growing substantially in order to effectively meet the information needs of ARTEP users. More specifically, schools must:

- prepare "improved" ARTEP documents which effectively combine "how to train" guidance with the "what to train" guidance traditionally provided by these documents;
- periodically revise ARTEP documents to reflect the continual changes in tactical doctrine associated with the ongoing process of Army force modernization.

The tools generally available to ARTEP developers (i.e., typewriters and stand-alone word processors) appear inadequate given the growing ARTEP development workload. Therefore, the purpose of the present research was to define the concept of a Computer-aided ARTEP Production System (CAPS) in preparation for the development of a CAPS prototype. This research was conducted under the sponsorship of the proponent for ARTEP development, the US Army Training Board (ATB).

Procedure:

Phase I of this effort was concerned with identifying the type of ARTEP development functions to be facilitated by a CAPS. The process of developing "improved" ARTEP documents was studied as it is defined in TRADOC Reg 310-2 (Test) and as it is practiced within the school designated to host the CAPS prototype, the US Army Infantry School (USAIS). The specific objectives of this phase were to:

- identify, describe and size the ARTEP-related products and interim products prepared by each USAIS organization involved in ARTEP development;
- identify, describe and size the source materials used in preparing each product;
- define decisions to be made when preparing each product.

In Phase II, computer technologies which have been applied to document preparation were identified and critiqued in terms of their applicability to the ARTEP development process defined in Phase I. A key issue was the type of data base management system (e.g., relational versus hierarchical) which is best suited to ARTEP development.

A concept of a CAPS was defined and illustrated in Phase III. Alternative lists of commercially available hardware/software compatible with the concept (i.e., and compatible with other hardware/software noted in the same list) were developed. Estimates of the research/development tasks necessary to implement the development of a CAPS prototype were prepared.

Findings:

- ARTEP preparation requires service schools to manage what might be viewed as three interrelated data bases --
    - tactical doctrine (e.g., FMs);
    - training requirements (unit missions, collective tasks and individual tasks to be trained);
    - training plans.
- A given item of information within one data base may bear a number of functional relationships to items within the other data bases and to other items within the same data base. A relational data base management system would provide a flexible and concise method of recording these relationships.
- Estimates of the amount of data to be stored within a CAPS suggest the need for a minicomputer.
- Approximately 50 percent of the information provided within improved ARTEP documents is in the form of figures and tables. Graphics terminals and a printer with a high degree of graphics capability appear to be required elements of a CAPS. Commercially available graphics terminals and printers meeting CAPS specifications were identified.
- Ideally, the CAPS should be a menu-driven system which guides users through the ARTEP development process. Such a system would reduce formal training requirements. The use of a relational data base management system as the core of a CAPS greatly facilitates the development of menus, because short, standardized queries and command statements (i.e., often a single line) replace traditional computer programming.
- The early phases of CAPS prototype development will be concerned with:
    - refining CAPS functional requirements (e.g., relating each decision made in preparing a product to specific source materials rather than relating the product as a whole to these materials);
    - designing the CAPS data base to effectively address refined CAPS functional requirements.

- writing standardized data base queries and command statements for extracting information required to support specific ARTEP developer decisions and for loading/changing data.

Utilization of Findings:

The findings of this effort will be considered in the selection of commercially available hardware and software for a prototype CAPS. These findings will also be employed in developing a prototype CAPS within USAIS.

CONCEPT STUDY OF THE COMPUTER-AIDED ARTEP PRODUCTION SYSTEM (CAPS)

CONTENTS

LIST OF FIGURES

LIST OF FIGURES (CONT'D)

LIST OF TABLES

CONCEPT STUDY OF THE COMPUTER-AIDED ARTEP PRODUCTION SYSTEM (CAPS)

INTRODUCTION

This report is the final technical report of a six month study sponsored by the Army Training Board and technically monitored by the Army Research Institute (ARI) under Contract No. DABT60-84-C-0109 to produce a conceptual design for a Computer-aided ARTEP Production System (CAPS).

The Problem

Army branch service schools prepare Army Training and Evaluation Program (ARTEP) documents as guides to unit training. The ARTEP consists of a multi-echelon, integrated, combined arms training program featuring unit missions and the collective tasks, conditions, and standards necessary for mission success. It includes structured evaluation outlines intended to provide units with data to formulate remedial training strategies.

The ARTEP development workload of each school has been increased by two events. First, changes in tactical doctrine associated with force modernization impose requirements to revise existing ARTEPs or to prepare ARTEPs for new types of units. Second, the ARTEP is being transformed from a document that merely describes unit training requiremens to one that provides decriptive unit training plans (TRADOC Regulation 310-2. Development, preparation, and management of Army Training and Evaluation Program, January, 1984). This transformation requires new types of analyses to be performed by ARTEP developers, and it requires the preparation of new types of products under the rubric of "ARTEP." Both force modernization and ARTEP improvement add to an already overburdened manual ARTEP production process.

Specific problems inherent in <u>any</u> paper and pencil system dealing with the number of variables addressed by the ARTEP are:

1. The ARTEP production process is not responsive to its users nor to its environment. As suggested above, when doctrine, tactics, and equipment are changed, the underlying tasks, conditions, and standards of performance for successful mission accomplishment will change as well. Not only must the ARTEP continue to support current battlefield TO&E, but also it must keep pace with the programmed introduction of new systems technology, (e.g. Force Modernization Products), and new organizational structures, (e.g. Division 86), as they are fielded. The current two year production cycle for the development/revision of the ARTEP cannot keep up with these developments, and thus new ARTEPs are frequently obsolete when issued.

2. The lengthy ARTEP production process also inevitably requires that the responsibility for authorship of specific content passes through two or three hands during the development/revision cycle. While general standards for style

and content do exist, it still allows for considerable latitude in quality and comprehensiveness. Thus, a document of uniform high quality is unlikely. Furthermore, variances in style and terminology result in misinterpretations by users.

3. ARTEP authors will normally be trainers or subject matter experts, not professional writers or instructional developers. Many may have difficulty in producing professional level training products, and all will require a period of familiarization to become aware of the requirements of the job and the resources at hand. The training investment required to create a good ARTEP author must be amortized over a relatively short period due to normal personnel rotation cycles. The current two year development time results in a wasteful diffusion of costly training as personnel are lost and new staff take their places.

4. Current ARTEP documents are, for the most part, too bulky to be conveniently useful. They include training programs for everyone, from combat arms personnel to fuel truck drivers. The result is that the document is difficult to carry to the field and too complex for quick reference. The broad scope of activities included guarantees that frequent revisions will be required, and as noted above, these take an inordinately long time to produce. When produced, the cost of revision, printing, and distribution is unreasonably high because the unrevised material must be reissued as well.

5. The cumbersome ARTEP production and distribution process precludes 1) ARTEP tailoring by the SME/authors to program the individual needs of local users, and 2) timely revisions of ARTEP's in response to feedback from users.

Taken together, these deficiencies inherent in a manual ARTEP production system prevent the ARTEP from realizing its full potential as a meaningful and effective training tool for the units. Until now, technology had little to offer to improve the situation when the constraints of cost and manpower support resources were considered. Recent developments in low-cost, easy-to-use distributed information processing systems, and progress in defining more precisely the process by which ARTEP missions and tasks are produced set the stage for implementation by interactive automation.

The Solution

It is apparent that what is required are new tools, techniques, and organizational approaches for the production of ARTEP that are (1) more efficient, (2) more responsive, and (3) less costly than the current manual approach. The technological bases for these developments are now realizable through exploitation of recent advances in:

- Distributed computational networks.
- Electronic information distribution.
- High performance intelligent workstations.
- Inexpensive very large scale mass storage.
- Artificial intelligence tools to assist non-expert human users.
- Advanced man-machine interaction protocols to relieve users of the burden of understanding system operation and facilitate concentration on content rather than process.

Computer-Aided ARTEP Production System (CAPS)

CAPS will take advantage of the technologies mentioned above to provide improved efficiency, responsiveness, and less costly production of ARTEP. It will support authors in the following areas:

- ARTEP database storage, query, and management.
- ARTEP authoring and revision.

CAPS will initially support these functions on a stand-alone basis, but when mature it is viewed as one component of a distributed training authoring, distribution, management, and evaluation network. Figure 1 presents a notional schematic of this mature system. The CAPS portion of this system is founded on the development doctrine that makes individual authors responsible for specific ARTEP component products. Change authority will be restricted to the designated sponsor for each section. Thus, the workstation topology will be structured to parallel lines of authority for authoring and updating responsibility. Although in our concept, any user can review any material resident in the system as needed, he can only revise those portions that are his assigned responsibilities. The importance of this structure is that as the system becomes aware of the need for ARTEP revision, work assignments can be routed automatically to appropriate authors.

Initially, when operated in its stand-alone configuration, CAPS will produce the ARTEP (or, to use the latest terminology, ARTEP Mission Training Plans and Drills) from a standard database maintained by subject matter experts, and distributed in paper form to end users. Eventually, when part of an electronic distribution network such as the one illustrated in Figure 1, a current ARTEP can be maintained in mass storage at the ARTEP development site, and can be distributed via low bandwidth links (modem net) or via mailed magnetic disks or tapes, as required to commanders/units. This will not only eliminate the need for large printed volumes in the field, but will also make available to field commanders electronic access to the entire ARTEP database. The commander can then use this information to tailor his ARTEP to specific mission and location requirements.

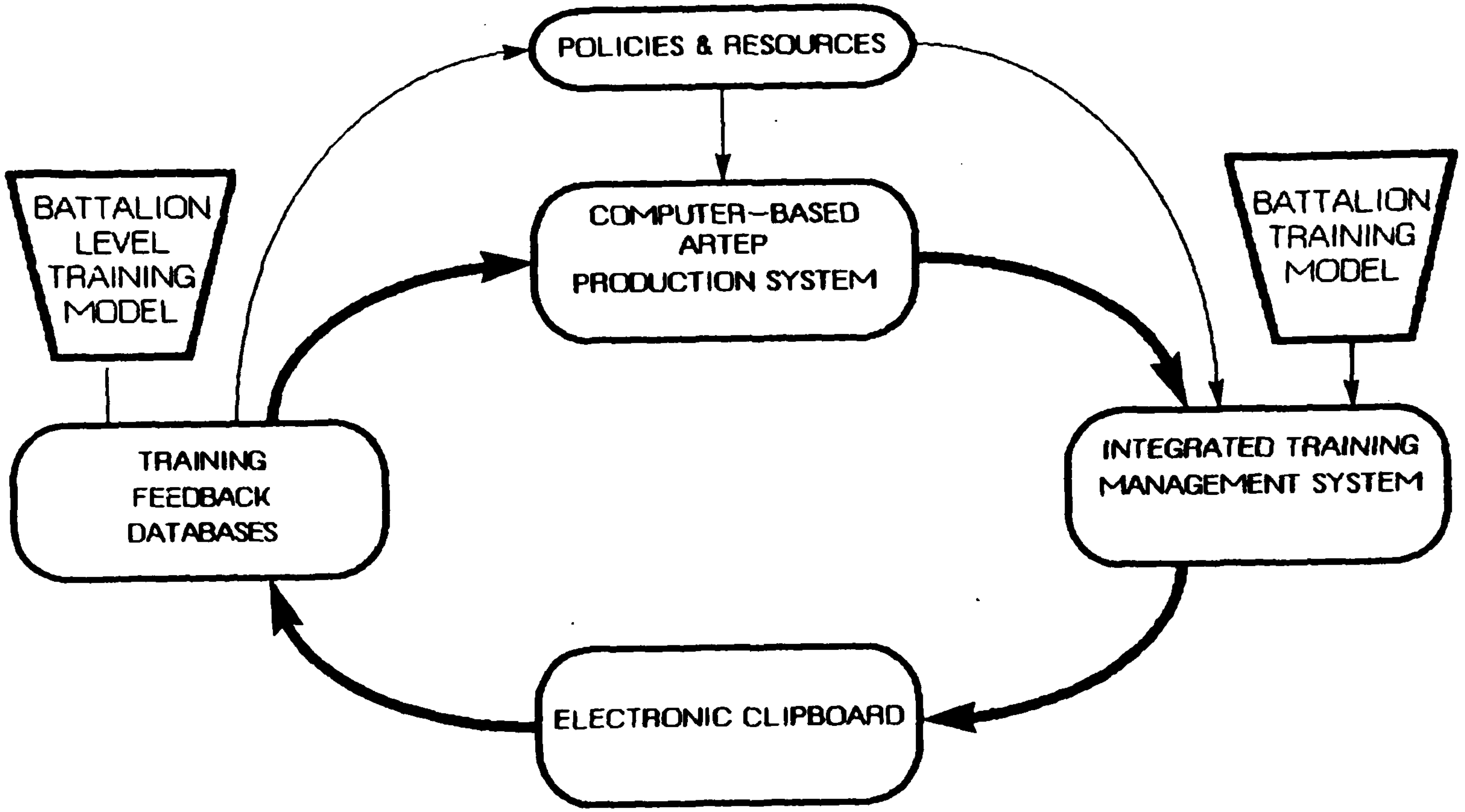

POLICIES & RESOURCES
BATTALION LEVEL TRAINING MODEL
COMPUTER–BASED ARTEP PRODUCTION SYSTEM
BATTALION TRAINING MODEL
TRAINING FEEDBACK DATABASES
INTEGRATED TRAINING MANAGEMENT SYSTEM
ELECTRONIC CLIPBOARD

FIGURE 1
ELECTRONIC UNIT TRAINING SYSTEM CONCEPT

Ideally, CAPS and the systems to which it is linked, will eventually become the mechanism for linking training resource expenditure to training proficiency. Such information is required to justify the acquisition of training resources and to support rational distribution of these resources.

In summary, the provision of shared data resources and information distribution throughout the ARTEP development process to support automated production, updating, and distribution of products to the field will:

- Produce a better product in a more timely manner (i.e., less subject to errors and more valid because the production process incorporates performance feedback as an integral part of the system).
- Establish a common database that is less sensitive to personnel turbulence within both the TRADOC school and the using unit.
- Increase the productivity of the ARTEP development team, who will also be required (in accordance with the latest TRADOC policy) to author subjects they teach.
- Increase responsiveness to changes in hardware, doctrine, trainer/soldier profile, training policies, procedures, and command guidance.
- Create a database that is useful for personnel selection, training evaluation of new systems, training resource acquisition and distribution in accordance with mission priorities, and that will provide a necessary database to support research in training technology.
- Provide training support packages rapidly to support contingency deployments.

Objectives of the Study and Contents of the Report

The objective of this study was to develop a concept and a design, to include new tools, techniques, and organizational approaches, for the production of ARTEP. The aim of the automated ARTEP production system will be to provide improved efficiency, responsiveness, and less costly production of ARTEP through the innovative use of interactive automation to assist in:

1. ARTEP database storage, query, and management
2. ARTEP authoring and revision.

Haphazard automation is potentially a serious problem resulting in wasteful duplication of effort and/or inadequate attempts to automate funtions. Army Regulations 18-1, Army Automation Management, was developed to guide the application of automation to Army needs. The present study was conducted to estimate the hardware/software requirements for developing a prototype CAPS within a designated school. The present study, combined with the development of a prototype, represents the concept development phase of a CAPS within the Army Automated System Life Cycle.

The second chapter and the Appendices describe in detail the current state-of-the-world with respect to ARTEP production. This chapter focuses specifically on procedures and organization at the US Army Infantry School (USAIS) and the US Army Armor Center (USAARMC). The ARTEP database, as it is derived and organized, is discussed, as are the various doctrinal publications and TRADOC regulations related to ARTEP preparation.*

The third chapter describes the functional requirements for the Computer-aided ARTEP Production System and analyzes the technologies that can be brought to bear in creating an effective CAPS system. The fourth chapter presents the details of the CAPS system design, including the database management system, the user interface, and the system architecture. The fifth chapter then presents our estimates of the program plan, the estimated manpower requirements, and the estimated hardware costs.

This study was conducted with the generous cooperation and assistance from the commanders and staff members of several TRADOC commands, including the US Army Training Board, the US Army Infantry School, the US Army Armor School, and the US Army Air Defense Artillery School.

*Chapter 2 refers to a number of Appendices relevant to the ARTEP development process. These Appendices are contained within an ARI Research Note entitled "Appendices to Computer-aided ARTEP Production System (CAPS) Concept Study".

ANALYSIS OF ARMY TRAINING PRODUCTS AND PROCEDURES

Introduction

The Army Training and Evaluation Program (ARTEP) is a collection of training guidance to commanders and trainers in the field that identifies (1) combat critical tasks, (2) realistic battlefield conditions under which the tasks must be performed, and (3) minimum standards of performance. The ARTEP also provides a framework for commanders and trainers to develop their own effective individual and unit training and evaluation programs. Finally, the ARTEP (1) provides guidance that facilitates evaluation of unit training programs and individual and collective training proficiency, and (2) facilitates identification of training deficiencies to allow development and conduct of remedial training.

While other portions of the ARTEP identify skills that must be trained, the ARTEP Mission Training Plan (AMTP) provides guidance to assist the trainer in determining how to train to specific standards within the ARTEP framework of a combined arms force in a tactical situation, i.e., Mission, Enemy, Troops, Terrain, and Time (METT-T).

Changes to the current ARTEP, embodied in TRADOC test regulations, intend the term "ARTEP" to encompass the entire body of Army training literature. The current ARTEP will be restructured and renamed the "ARTEP Mission Training Plan." In keeping with this "new" ARTEP, this report will use the term "Mission Training Plan," or "MTP," to refer to the collection of training guidance that is produced for the field commanders.

The purpose of this chapter is to describe the MTP production process. This process is composed of two distinct developmental efforts, as follows:

(1) Extraction of the tasks that must be accomplished to fight effectively from "How-to-Fight" doctrine.

(2) Incorporation of these tasks into "How-to-Train" programs that, when executed by commanders and trainers in the units, will ensure the "How-to-Fight" capability.

A Systems Approach To Training

Overview. TRADOC Regulation 350-7 (DRAFT), A Systems Approach to Training, dated 11 January 1984, prescribes TRADOC policy, requirements, and responsibilities for the systematic analysis, design, development, and implementation of training programs and support materials. This regulation presents a model for a systems approach to training. Along with TRADOC Regulation 310-2 (Test), Development, Preparation, and Management of Army Training and Evaluation Program, it establishes the Army standard as to MTP format, contents, and standardization.

To understand the support requirements of an MTP author, however, the elements of both TRADOC Regulation 350-7 (Draft) and Regulation 310-2 (Test) must be viewed in terms of their interrelationships. This section will comprise a walk through the world of the MTP author which will, in conjunction with the appendices that are cited, describe these relationships and the manner in which an MTP is created. To begin, consider the Platoon Mission Training Plan illustrated in Figure 2.

The illustrative platoon Mission Training Plan (MTP) in Figure 2 is a greatly simplified depiction of how individual soldier actions, planned and controlled by squad and platoon leader actions, result in the accomplishment of unit (collective) tasks in the context of a mission. Here we depict the mission of "Hasty Attack." (See Appendix E, Glossary, for a definition of Missions, Tasks, etc.) The Hasty Attack Mission will be used for illustrative purposes throughout this chapter. The MTP is a product of the appropriate TRADOC School. The purpose of the MTP is, as stated above, to "package" the doctrine of "How-to-Fight" into guidance on "How-to-Train." The complex interrelationships of tasks to missions, which comprise the heart of the Mission Training Plan, is not simply the result of the MTP developer's creativity. Rather, it emerges from a systematic investigation of requirements that begins with what is called a "collective front-end analysis" and ends with an evaluation of the training programs themselves, measured in terms of cost and effectiveness. In examining this process let us begin at the beginning.

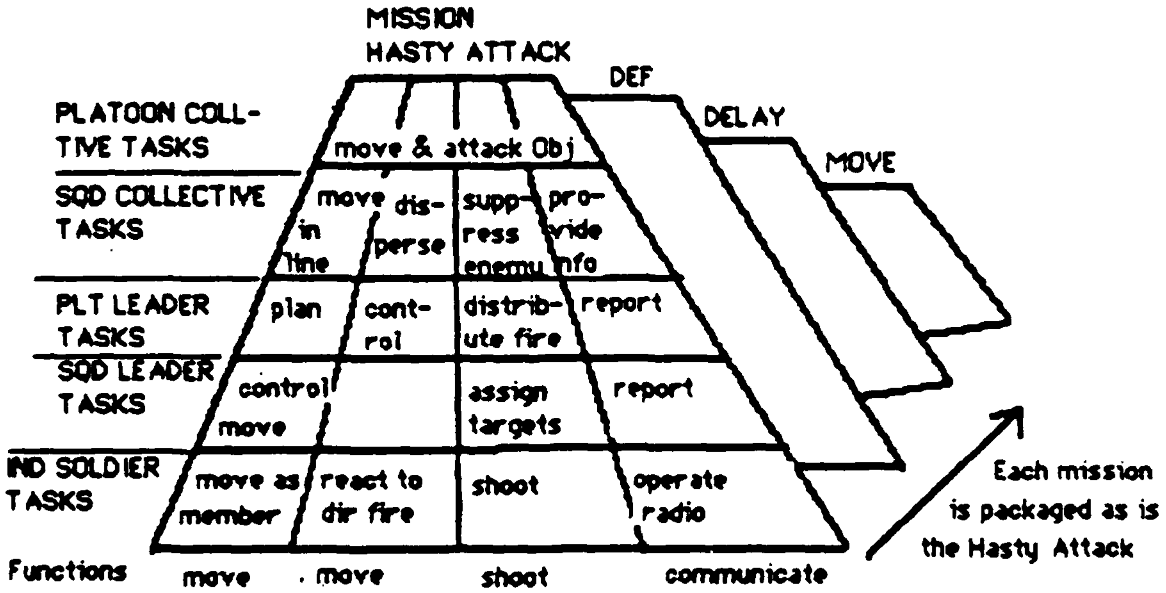
Collective Front-End Analysis. The process of Collective Front-end Analysis (CFEA) begins at the TRADOC Proponent School or Integrating Center when one of the following "triggers" occurs:

1. New equipment is adopted/developed (i.e., the M1 Tank).
2. Modified equipment is adopted/developed (i.e., the M60A3 Tank).
3. A new organization is created or modified.
4. New doctrine or "How-to-Fight" concept is adopted.
5. A training shortfall is identified.

Analysis includes an examination of the threat, doctrine, organization, and equipment. Critical individual tasks are derived from collective tasks which, in turn, were identified from the unit missions. The US Army Armor School's procedure for the conduct for the collective front-end analysis is described in Appendix A. The Armor School's CFEA procedures have been in use since 1982 and comply generally with TRADOC Regulation 350-7 requirements. For these reasons, and because the Armor School is a co-author of ARTEP 71-2, this CFEA is used as an exemplar throughout. A summary of the procedures employed is provided below.

The front-end analysis process encompasses five (5) phases. (Readers must turn to Appendix A for detailed data.)

Phase 1. Unit Familiarization. During this phase the analyst collects source material, analyzes that material and produces a data package detailing the missions, capabilities and organization of the unit. (For the



PLATOON COLL-TIVE TASKS	move & attack Obj			
SQD COLLECTIVE TASKS	move in line	disperse	suppress enemy	provide info
PLT LEADER TASKS	plan	control	distribute fire	report
SQD LEADER TASKS	control move		assign targets	report
IND SOLDIER TASKS	move as member	react to dir fire	shoot	operate radio
Functions	move	move	shoot	communicate

Each mission is packaged as is the Hasty Attack

FIGURE 2. Platoon Mission Training Plan

CAPS study two units are to be examined: The Mechanized Infantry Battalion equipped w/the M2 Bradley Fighting Vehicle and the Tank Battalion equipped with the M1 Main Battle Tank.)

Phase 2. Mission Analysis. This phase identifies the unit's stated and implied missions. It is at this point that information is produced that will be used to generate ARTEP missions. Operational mission diagrams for each echelon are prepared (i.e., squad/platoon/company and battalion). Unit missions are stated in the unit's Table of Organization and Equipment (TO&E), a document prepared by DA that authorizes the Army to man and equip the type unit. Implied missions, however, cannot be found on the TO&E and must be derived by analysis of other data sources, such as doctrinal manuals. Analysis of missions produces the type data illustrated in Table 1.

Phase 3. Collective Task Identification. In this phase the analyst identifies and records the collective tasks which must be performed, by each echelon, to execute the missions identified in Phase II. Three types of collective tasks are developed and two products result from this phase. The task types include (1) operational collective tasks, (2) sustaining collective tasks, and (3) equipment specific, hardware-driven, collective tasks. Products from this phase available to the ARTEP author are (1) a mission-to-collective task matrix, and (2) the initial collective task inventory. Examples of the process and products from this phase are provided in Appendices A and B.

Phase 4. Verifying the Task Inventory. The purpose of this phase is to verify the initial task inventory developed in Phase 3 and to uncover/develop additional collective tasks that may have been omitted during the analysis. Verification is conducted by the use of questionnaires and/or surveys of subject matter experts (SME) and field units.

Phase 5. Collective Front-End Analysis. This phase accomplishes the following:

(1) Identifies proponency for each task (i.e., the Infantry, Engineers, Armor, etc.),

(2) Analyzes conditions under which the task will be performed and selects those tasks that most accurately represent the combat environment in which the unit must perform the task.

(3) Determines minimum performance acceptable for combat success under selected conditions.

(4) Verifies that performance standards are measurable, reliable, and valid for all units.

(5) Identifies and records collective, individual, and leader task relationships.

(6) Produces (a) task documentation and (b) individual/collective task integration.

TABLE 1

MISSIONS PERFORMED BY ECHELONS

Functions:	MOVE				OFFENSE				DEFENSE			
Echelon:		Bn	Co	Plt		Bn	Co	Plt		Bn	Co	Plt
Missions:												
	Movement to Contact	X	X	X	Hasty Attack	X	X	X	Delay	X	X	X
	March	X	X	X	Deliberate Attack	X	X	X	Defend in in Sector	X	X	X
	With-drawal	X	X	X	Recon in Force	X	X	-	Occupy a Battle Position	X	X	X
					Deliberate River Cross.	X	X	-				

In addition to the above, the CFEA process develops, for each collective task, the following information: (1) task elements (i.e., the subtasks that comprise the collective task), (2) the component skills and knowledges required for execution of the task element, and (3) collective task standards and references. Clarifications and special considerations that must be considered when developing training programs and/or executing the collective task are also provided.

Task Documentation Example. Table 2 provides an illustrative example of the product derived from the CFEA process.

Doctrinal and Individual Job & Task Analysis Input to the ARTEP Database at the Infantry School. Before proceeding further, it is necessary to acknowledge that although the TRADOC regulations require a CFEA, and while it is obvious that such a process and its products are beneficial to the development of a comprehensive MTP, time and manpower restrictions often combine with the needs of the field to force the MTP author to proceed without a formal, completed, CFEA. Indeed, our research reveals this situation to be the norm. How then do the schools proceed?

As will be shown later shown later in Figure 7, the Combined Arms Tactics and Doctrine Directorate (CATD), in concert with the Analysis and Studies Office and the Manuals and Test Branch of the Directorate of Training and Doctrine (DOTD), provide MTP authors with lists of missions (prepared by the authors of the doctrinal field manuals and instructors of the doctrinal and tactics classes taught at the school). Individual tasks are derived from the Soldier's Manuals and Skill Qualification Tests (SQT) developed by the Manuals and Test Branch of DOTD. This is not as haphazard as the casual reader might suppose. First of all, the doctrinal manuals and classes are reviewed by branch-qualified "murder boards" and the Soldier's Manuals and SQT are validated by units in the field. Secondly, doctrine, and thus missions and collective tasks, are reviewed by the units in the field and students at the service schools.

From this we can see that the absence of a formal CFEA causes problems of completeness rather than problems of accuracy and validity. Without a valid CFEA it is very difficult to ensure that every mission and collective and individual task is included in all the appropriate T&EOs and Mission Training Plans that form the heart of the ARTEP. For the CAPS study team, however, it is sufficient at this time to identify the hierarchical and lateral relationships of each type of data that is required so that a proper database and DBMS may be designed that will interface with the ARTEP author's existing data sources and will "get smart" as it is employed and updated by the analysts and doctrinal writers.

Task Relationships in Army Training

Multi-Dimensional Relationships Among Tasks. A way to look at the ARTEP database is to consider the vertical and horizontal relationships within the data. The interrelationships among ARTEP missions and collective tasks, individual leader, command, and command staff tasks are illustrated by the

TABLE 2

TASK IDENTIFICATION DATA

Echelons:	Scout Platoon
Task Statement:	Movement to Contact
Missions/Operations:	All missions under move, offensive, defensive, security, and reconnaissance operations.
Functional Areas:	Force Movement.
TO&E Unit:	Tank Bn, Mech Inf Bn, Divisional Cavalry Troop/Squadron & Cavalry Regiment
Information Sources:	a. FM 17-95 Cavalry b. FM 71-1 & 71-2, Oct 81, The Abrams Battalion

Related Task:

	COLLECTIVE:	ECHELON	TASK ELEMENT
#129	Execute terrain driving	Platoon	6, 7 & 9
#126	Occupy turret down position	Crew	3
#127	Occupy hull down position	Crew	3, 4
#181	Establish all-round security	Platoon	3

	INDIVIDUAL	DUTY POSITION	TASK ELEMENT
#420	Direct Employment of Sct Plt 071-326-0608/	Platoon Leader	2
MQS #107	Communicate using visual techniques	Vehicle Commander	6
#371	Select movement route	Platoon leader/Sgt	2

cube in Figure 3, where the collective task of "Move Dismounted" is shown within the context of the ARTEP mission of "Hasty Attack." Down the right side of the cube is the ARTEP mission of "Attack" with the major variants "Hasty," "Deliberate," and "Night." Across the top are the echelons of command (squad thru battalion), and down the left side are the categories of tasks in terms of who performs them.

Look at one element of the data in Figure 3, that of the individual rifleman, moving as part of a dismounted squad moving to make contact with the enemy as part of the platoon conducting a "Hasty Attack." Figure 4 portrays each individual action within the context of the collective actions being taken by the squad.

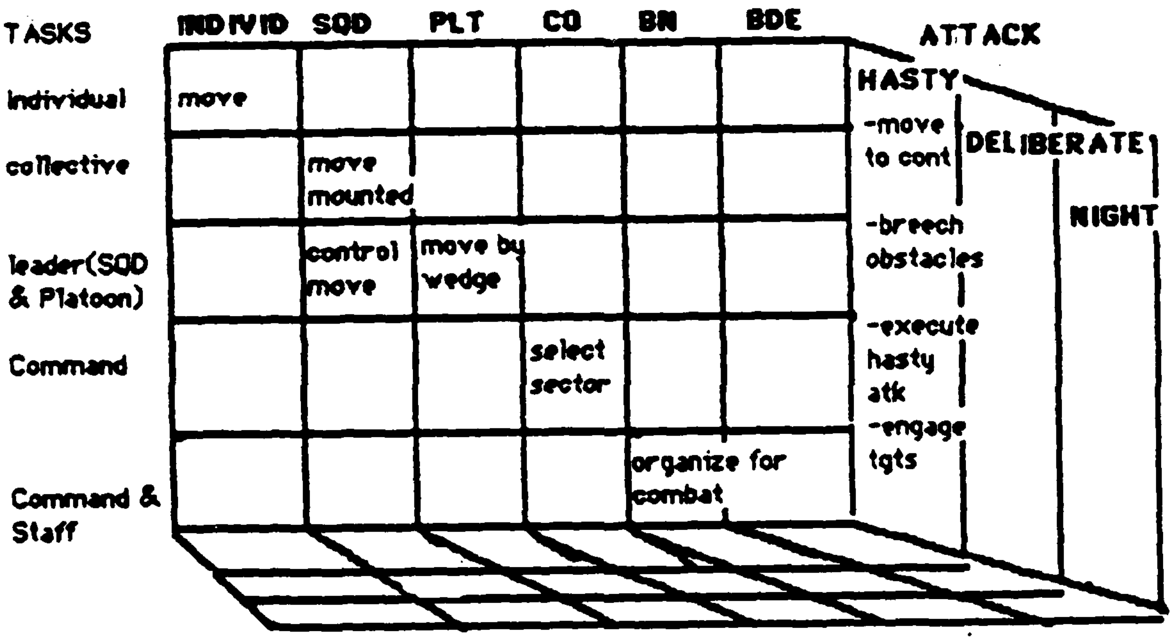
The contents and arrangement of the sub-cubes in Figure 3 are not invented by the MTP author. Rather each sub-cube represents a task laid out in Army doctrinal "How-to-Fight" Field Manuals and individual Soldier's Manuals. From these documents all of the tasks that must be performed by individuals and units in combat can be identified through the CFEA process, along with the standards to which they must be performed.

The cube analogy provides a logical view of the ARTEP database from the MTP user's (author's) point of view. However, it does not totally capture the interrelationships of the information elements in that database. Another way of conceptualizing the database is to think of it as a many-dimensional matrix, with axes labelled as follows:

a. Mission(s)
b. Mission Variant(s)
c. Task(s)
d. Performer(s) (individual(s))
e. Echelon(s) (e.g., individual soldier/officer thru battalion)
f. Drill(s)
g. Situational Training Exercise(s)
h. Field Training Exercise(s)
i. Mission Training Plan(s)

Although it is difficult for most individuals to visualize an n-dimensional matrix, it is important that CAPS developers be sensitive to the fact that the relationships among information elements of the ARTEP system are not simply hierarchical, as simple collective-individual task integration might suggest. To help visualize the relationships between ARTEP Missions, Mission Variants, Tasks, Task performers (both individuals and units), and Echelons, a portion of the n-dimensional matrix has been shown as a cube in Figures 3 and 4.

Individual and Collective Tasks. An ARTEP Mission, which forms the basis of the MTP, is made up of individual soldier tasks, individual leader tasks, squad/crew and platoon collective tasks. Command tasks at the company level and command and staff tasks at the battalion level are added to encompass the universe of missions and tasks that a battalion-sized unit is expected to execute in combat. When described in doctrinal/tactic field manuals, these missions and tasks, in the aggregate, describe how the Army fights. The MTP describes how to train to fight. The problem faced by the MTP author is how



TASKS	INDIVID	SQD	PLT	CO	BN	BDE
Individual	move					
collective		move mounted				
leader(SQD & Platoon)		control move	move by wedge			
Command				select sector		
Command & Staff					organize for combat	

FIGURE 3. Individual Task-to-Mission-to-Echelon Integration

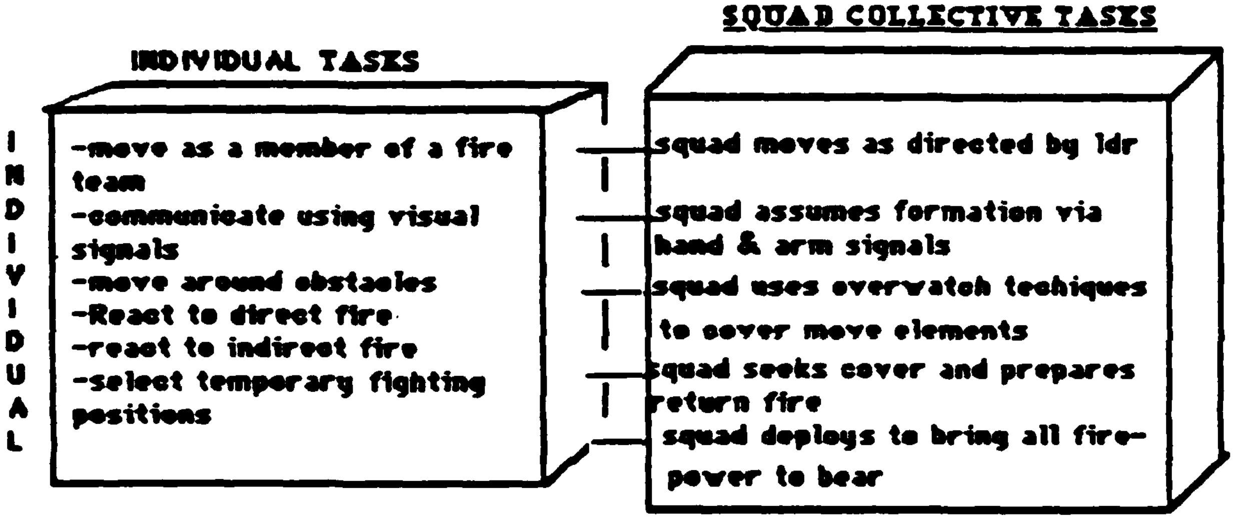

SQUAD COLLECTIVE TASKS
INDIVIDUAL TASKS
I
N
D
I
V
I
D
U
A
L
-move as a member of a fire team
-communicate using visual signals
-move around obstacles
-React to direct fire
-react to indirect fire
-select temporary fighting positions
squad moves as directed by ldr
squad assumes formation via hand & arm signals
squad uses overwatch techiques to cover move elements
squad seeks cover and prepares return fire
squad deploys to bring all fire-power to bear

FIGURE 4. Individual-Collective Task Integration

to ensure that training programs develop the capability to train as we intend to fight. The Assistant Commandant of the United States Army Infantry School, Brigadier General Edwin Burba, Jr., has outlined the task relationships with which the MTP developer must concern himself (shown in Table 3).

The ARTEP Data Base

All of the information derived during the CFEA, including the relationships between information elements, can be thought of as part of a database. The other part of this database is the information that MTP authors have previously created and included in it (e.g., previously-created ARTEP documents, etc.). From this database, the training program is produced by sorting related items by various attributes and adding new items when the sorting reveals insufficient information in the database.

Organization and Communication of ARTEP Data. The US Army Infantry School organization for ARTEP database development is shown in Figure 5.

We have said that the information necessary to create the complete version of the cube, illustrated previously in Figure 3, is derived from the CFEA. However, the conduct of a CFEA also requires input from the doctrine developers. As shown in Figure 5, Infantry doctrine is written by members of the school's Combined Arms Tactics and Doctrine (CATD) Department. Doctrine is promulgated through the publication of "How-to-Fight" Field Manuals (FM) for purposes of tactics instruction. Conduct of the CFEA is the responsibility of the Analysis and Studies Office of the school's Training and Doctrine Directorate. Briefly, the Analysis and Studies Office conducts an analysis of the CATD developed doctrine and produces the task documentation summaries which define the universe of tasks which must be trained to develop unit capabilities to perform assigned missions to defined standards. Figure 5 depicts a database of interrelated echelons (individuals through brigade), missions for each echelon and individual and collective tasks as discussed previously in Table 3. The cube shows the relationships between the mission and tasks and the performers. The information is organized logically; i.e., in the way in which a user can visualize the data being stored. The physical storage of data in memory may be different, but the data in the cube is the doctrine. Also the data defines the tasks to be trained to ensure that the Army "trains as it is to fight."

Once the information is so organized, the MTP author's task becomes that of determining the proper sequence of training to ensure (1) that precursive individual tasks are trained before collective tasks which incorporate them, and (2) that collective tasks are trained in drills before an attempt is made to train them in a mission (situational) context. Figure 6 depicts one "slice" of an MTP containing all elements of a doctrinally preferred Mission Training Plan for the individual to the brigade.

When addressing this database, it is useful to describe the fundamental role of knowledge in distributed environments (after all, the data and their complex interrelationships, within the cube, are really a body of knowledge that we wish to make available to each MTP author on demand). An individual author in this development system depends on his knowledge to drive his

Table 3

RELATIONSHIPS AMONG ARMY TASKS

"In a hasty attack or within a 'Move' battledrill, for example, soldiers do Skill Level 1 & 2 Soldier's Manual Tasks, i.e.,

- Move as a member of a fire team
- Move thru direct fire
- React to direct fire
- etc.

"while squad leaders do Skill Level 3 Soldier's Manual tasks, i.e.,

- Control fire team movements
- Select routes for armored vehicles
- Select overwatch positions
- Use visual Signals to Control Movement
- etc.

"while platoon leaders do Skill Level 4 Soldier's Manual tasks, i.e.,

- Control squad movements
- Implement movement techniques -- traveling overwatch and bounding overwatch
- Prepare and issue orders
- Control fires
- Report
- etc.

"Squads perform collective tasks per doctrinal FM 7-7, i.e.,

- Move dismounted
- Move mounted
- Recon woodline
- React to direct fire & air attack
- etc.

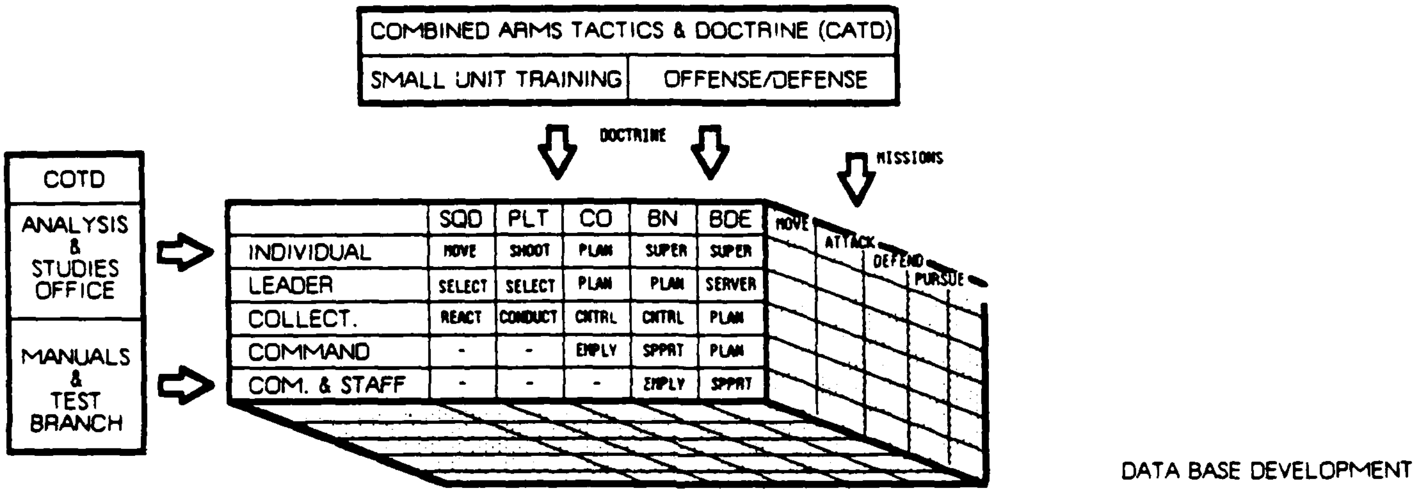
"Platoons perform collective tasks per FM 7-7, i.e.,

- Conduct suppressive fire
- Bypass
- Break contact
- Close on objective mounted-assaulted dismounted
- etc."

Table 3 (cont.)

"Commanders and staffs at company, battalion, and above (1) analyze threats and terrain, (2) organize for combat, (3) select formations, (4) select routes of advance and employ combined arms, (5) provide administration and logistical support, (6) and command and control all elements.

"Tasks in this vertical hierarchy (individual thru battalion command) also fit naturally into a horizontal relationship (e.g., move, shoot, communicate, etc.). When addressed in combination with the vertical components, this natural aggregation of soldier, leader, command and staff, and collective tasks into these horizontal functional areas is significant because it provides clear and concise means to train to criteria. Importantly, as you go from one ARTEP Mission to another, commanders and staffs do remarkably different tasks, but platoons, squads, and their leaders and soldiers do virtually the same thing."



	SQD	PLT	CO	BN	BDE
INDIVIDUAL	MOVE	SHOOT	PLAN	SUPER	SUPER
LEADER	SELECT	SELECT	PLAN	PLAN	SERVER
COLLECT.	REACT	CONDUCT	CNTRL	CNTRL	PLAN
COMMAND	-	-	EMPLY	SPPRT	PLAN
COM. & STAFF	-	-	-	EMPLY	SPPRT

FIGURE 5. US Army Infantry School Database Development Organization

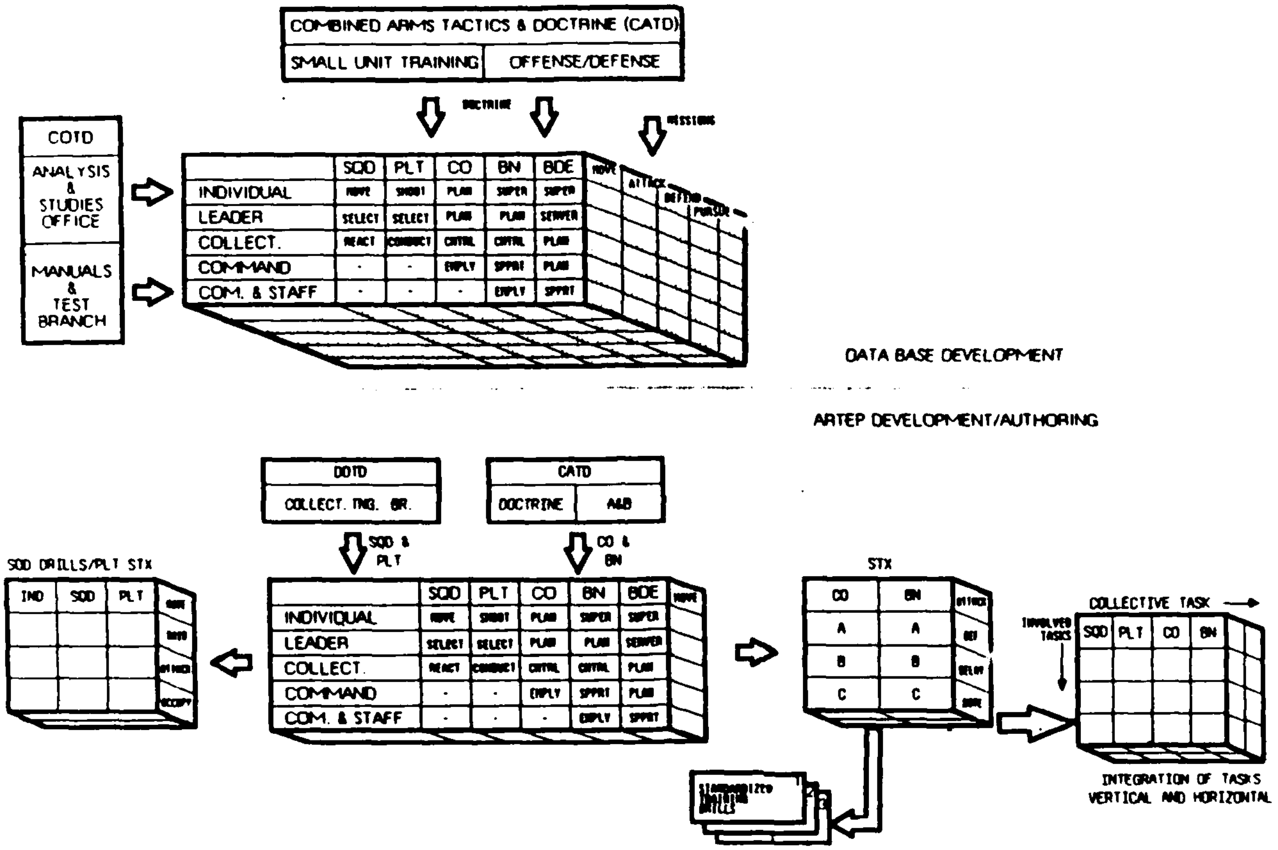


FIGURE 6. ARTEP Development/Authoring

decisions and actions. When individuals' actions have an effect on one another, it is often necessary that their actions be coordinated. This coordination is best achieved by a combination of predetermined, common procedures (a plan). In this case, the plan consists of TRADOC Regulation 310-2 (TEST) and 350-7 (Draft) ment oned earlier.

Coordination also demands efficient communications to expand and refine the plan or product. The amounts, type, and format of information that are relevant or necessary to allow individuals to successfully carry out their individual parts of the plan vary greatly according to the nature of the dependence of their plans on the actions of others. The ARTEP, by its very nature, introduces a hierarchy of information requirements on the part of authors. Since the strongest notion of knowledge that a group can have is common knowledge, any ARTEP/MTP production system, automated or manual, must share knowledge as it is acquired or required. However, common knowledge is really not practically attainable in manual systems.

The ARTEP development and authoring at the Infantry School, illustrated in Figure 6, shows why this common knowledge is not practically attainable in a manual system. The relationships between MTP authoring, shown in Figure 6, is such that the variables are difficult to control. Different people, at different places, and at different times are injecting changes at various points in the work flow which are not available to other participants in time to influence their actions. As a result, a comparison of tasks addressed in the "How-to-Train" manuals against tasks required in the "How-to-Fight" manuals reveals a pattern of omissions and duplications. Given the complex interactions depicted in Figures 5 and 6, and the MTP development problems and requirements discussed above, this is not surprising.

Figure 7 illustrates the entire Mission Training Plan (MTP) production process from database development to ARTEP development and authoring through to the production of a Mission Training Plan for each unit echelon. NOTE: The triangles at the bottom of Figure 7 is a schematic representation of an MTP to illustrate how doctrinal and training developments result in unit training plans.

Analysis of the "New" ARTEP/MTP. Discussions with subject matter experts and ARTEP/MTP authors at the US Army TRADOC agencies revealed that changes envisioned in TRADOC Reg 310-2 (Test) will soon result in substantial and significant changes in the ARTEP content and format now used in ARTEP 71-2, (the ARTEP chosen by the Infantry School as the focus for the CAPS program). Figure 8 below depicts the form and content of the "new" ARTEP. The reader is reminded that the term ARTEP (Army Training and Evaluation Program) will soon come to mean the entire body of training literature and the current ARTEP,, as restructured in Figure 8, will be known as the Mission Training Plan (MTP). The data structures and work flow depicted previously in Figures 5 through 7 reflect production of documents in these new formats.

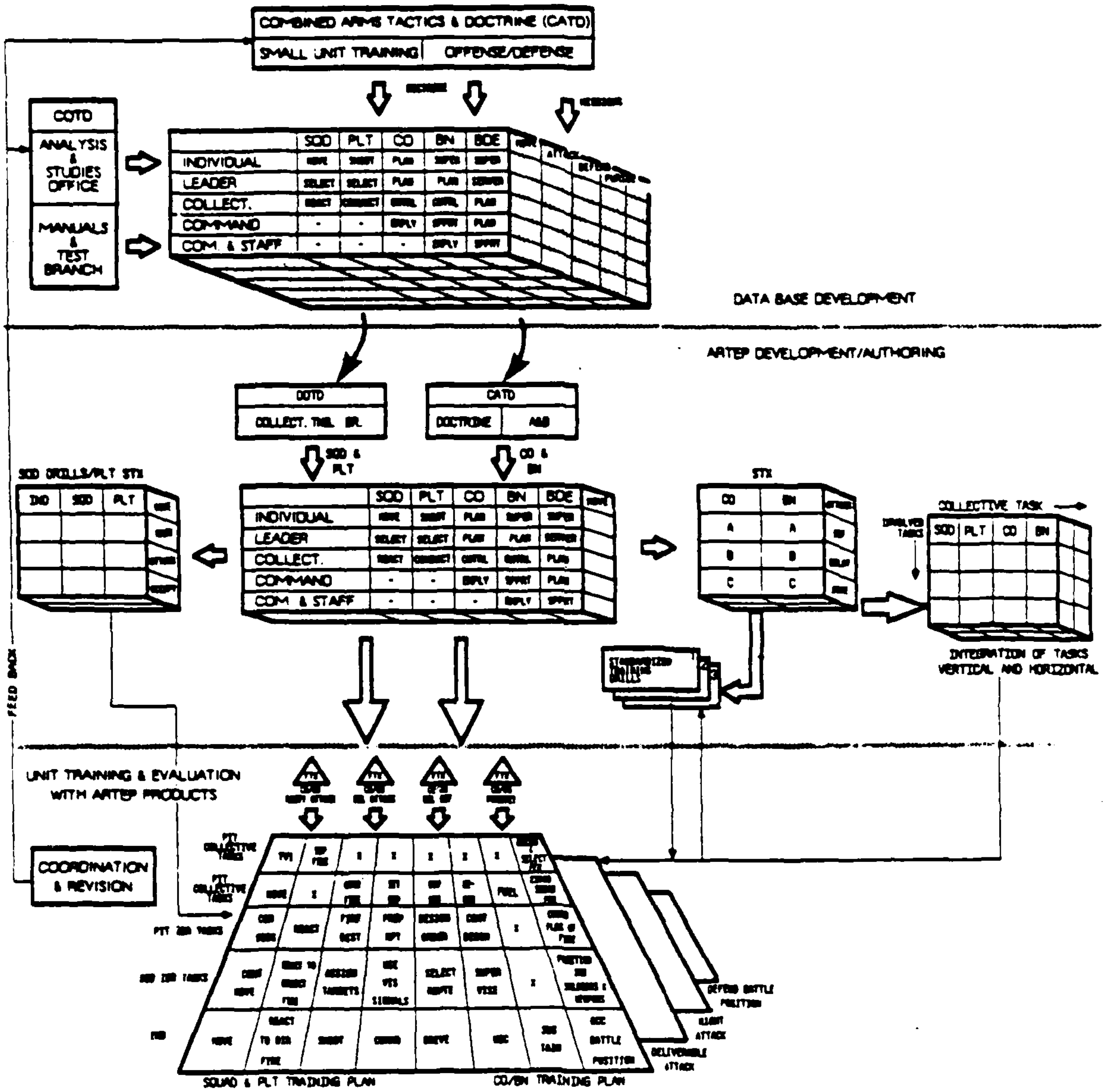


FIGURE 7. ARTEP Database Workflow at the US Army Infantry School

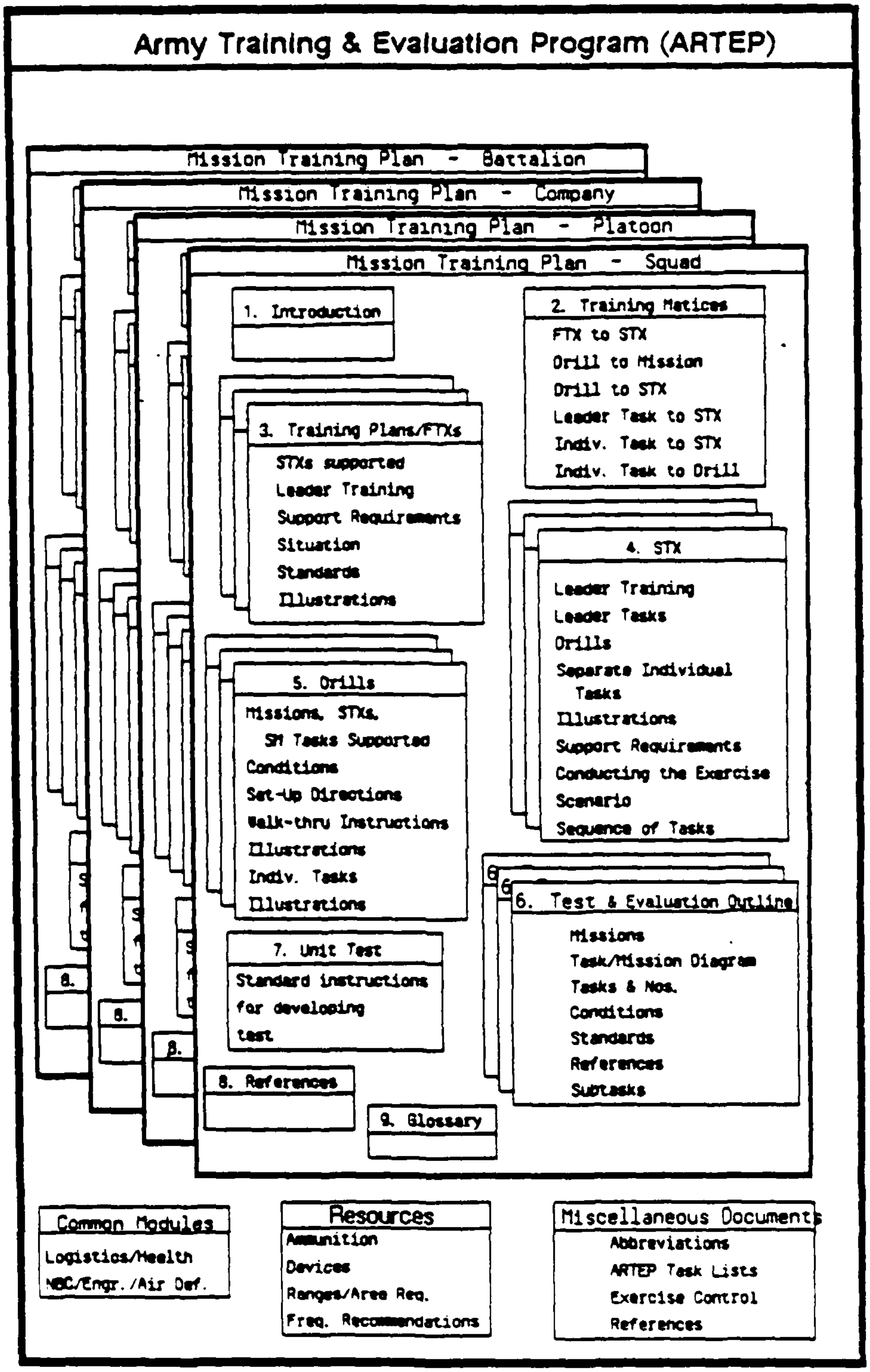


FIGURE 8. "NEW" ARTEP Contents

Development of a Mission Training Plan

MTP Authoring Procedures. TRADOC Regulation 310-2 (TEST) provides specific guidance as to the content, purpose, and format required for each section of the MTP. Figures 3 and 4, above, describe in general terms the data flow and products of the MTP and responsibilities for the creation and maintenance of the database at the Infantry School. What is missing are the detailed steps and decisions that each MTP author must make to turn "How-to-Fight" doctrine, missions, and tasks into effective "Train-to-Fight" programs. It is this process that is essential to capture within the "Expert Consultant" automation system known as CAPS. To begin, let us develop the hierarchy of MTP production processes within a framework of input-process-output. This methodology provides a graphical description of the functions performed by the system. Because we will describe the functions of the MTP production system and not its organization and logic, the reader will be provided with an overview of what the system does as an aid to understanding subsequent discussions of implementation (who does) unique to the Infantry School. As an aside, Appendix D contains a complete description of the USAIS workstation Data Flow in matrix form. This is essentially the same system description contained in this section, except that it displays each function of the system within the context of the Infantry School's organizational responsibilities.

Figure 9 depicts the hierarchical functions of the ARTEP production process of which the MTP production process is but a part. Note that each function is arranged so as to depict its relation to each other function.

Looking at the Collective Front-End Analysis (CFEA) function 3.0, we can see that it consists of two subordinate functions listed in blocks 3.1 and 3.2 of Figure 10. Each function of the CFEA process is further decomposed in Figures 11 and 12.

Figure 13 takes us from the CFEA/doctrinal development functions into the production of the MTP itself. The reader is urged to familiarize himself with the system's functions by scanning each step in the process. Figures 14 through 21 describe each major function in the process and relate each process to an output. The matrices in Appendix D depict these products in terms of which agency at the Infantry School produces the product and who uses it. The reader should be aware that a continuation of the input-process-output systems analysis can be made for each "process" identified above.

To continue, let us now examine some of the generic-type decisions that must be made. These are listed below as task criticality dimensions:

1. Learning difficulty. Is the task difficult to learn? Knowing how difficult in terms of time, resources, and performance is necessary to determine whether the task should be taught individually, collectively or in combination.

2. Performance difficulty. How difficult is the task to perform consistently to criteria? Does time since last successful performance influence difficulty?

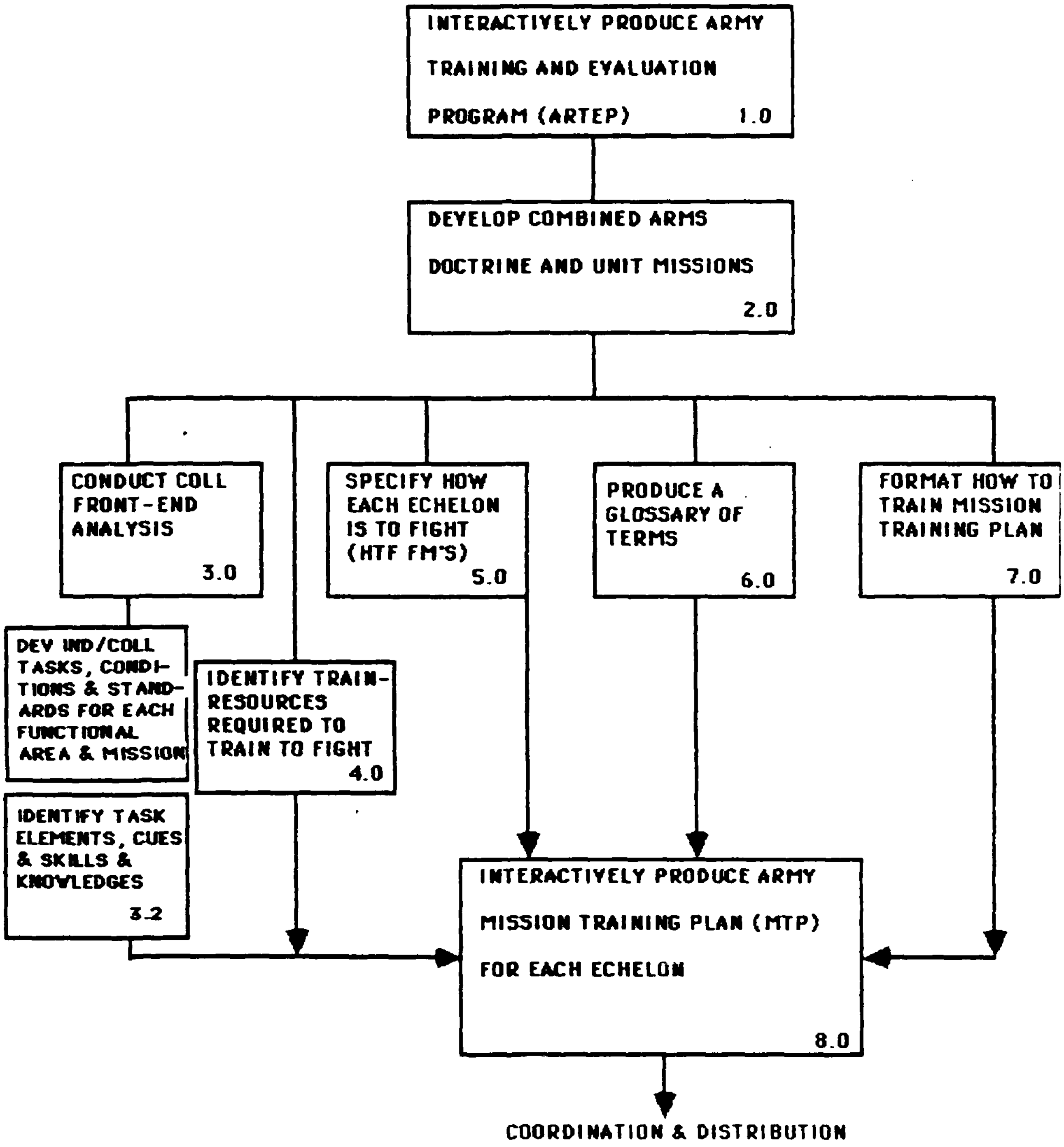

INTERACTIVELY PRODUCE ARMY TRAINING AND EVALUATION PROGRAM (ARTEP) 1.0
DEVELOP COMBINED ARMS DOCTRINE AND UNIT MISSIONS 2.0
CONDUCT COLL FRONT-END ANALYSIS 3.0
SPECIFY HOW EACH ECHELON IS TO FIGHT (HTF FM'S) 5.0
PRODUCE A GLOSSARY OF TERMS 6.0
FORMAT HOW TO TRAIN MISSION TRAINING PLAN 7.0
DEV IND/COLL TASKS, CONDITIONS & STANDARDS FOR EACH FUNCTIONAL AREA & MISSION
IDENTIFY TRAIN-RESOURCES REQUIRED TO TRAIN TO FIGHT 4.0
IDENTIFY TASK ELEMENTS, CUES & SKILLS & KNOWLEDGES 3.2
INTERACTIVELY PRODUCE ARMY MISSION TRAINING PLAN (MTP) FOR EACH ECHELON 8.0
COORDINATION & DISTRIBUTION

FIGURE 9. Hierarchy Plus Input-Process-Output System Analysis for CAPS

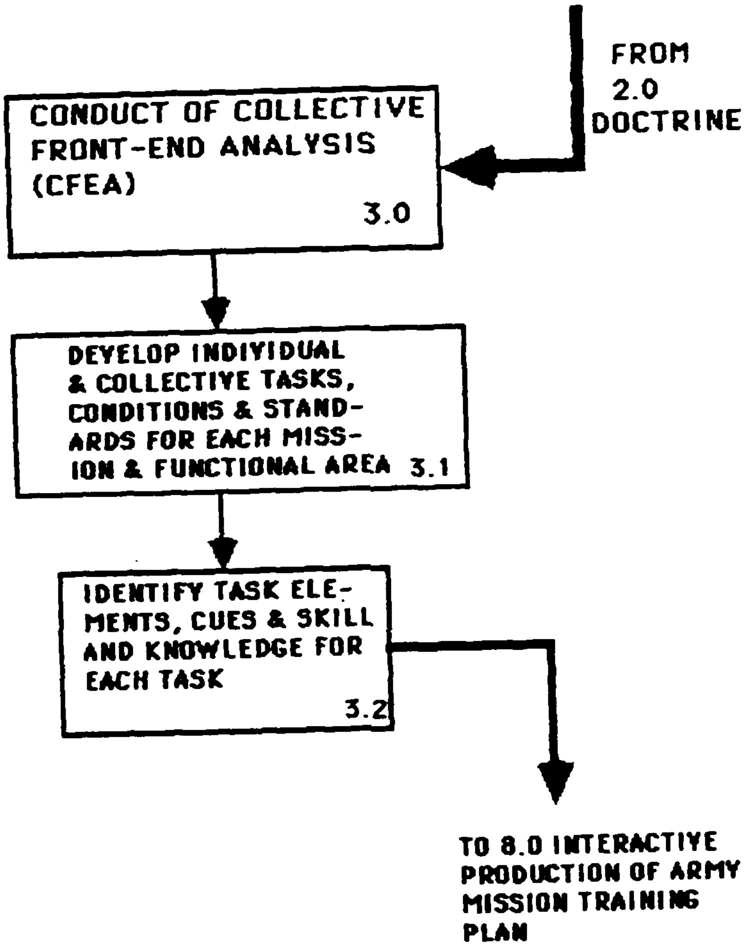
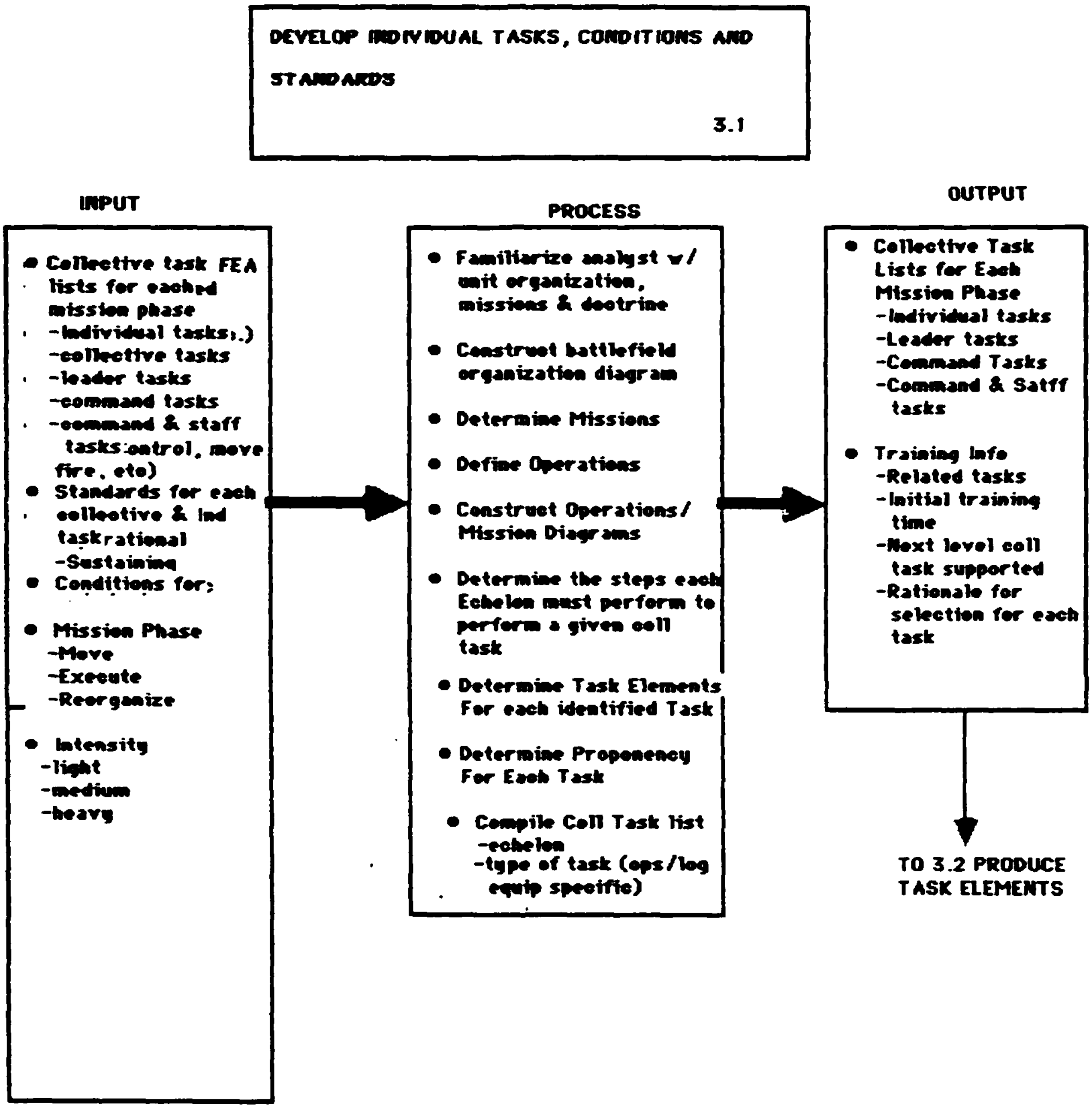

FROM
2.0
DOCTRINE
CONDUCT OF COLLECTIVE FRONT-END ANALYSIS (CFEA)
3.0
DEVELOP INDIVIDUAL & COLLECTIVE TASKS, CONDITIONS & STANDARDS FOR EACH MISSION & FUNCTIONAL AREA
3.1
IDENTIFY TASK ELEMENTS, CUES & SKILL AND KNOWLEDGE FOR EACH TASK
3.2
TO 8.0 INTERACTIVE PRODUCTION OF ARMY MISSION TRAINING PLAN

FIGURE 10. CFEA Hierarchy



DEVELOP INDIVIDUAL TASKS, CONDITIONS AND STANDARDS

3.1

INPUT

- Collective task FEA lists for eached mission phase
  - -individual tasks;.)
  - -collective tasks
  - -leader tasks
  - -command tasks
  - -command & staff tasks:ontrol, move fire, etc)
- Standards for each collective & ind taskrational
  - -Sustaining
- Conditions for:
- Mission Phase
  - -Move
  - -Execute
  - -Reorganize
- Intensity
  - -light
  - -medium
  - -heavy

PROCESS

- Familiarize analyst w/ unit organization, missions & doctrine
- Construct battlefield organization diagram
- Determine Missions
- Define Operations
- Construct Operations/ Mission Diagrams
- Determine the steps each Echelon must perform to perform a given coll task
- Determine Task Elements For each identified Task
- Determine Proponency For Each Task
- Compile Coll Task list
  - -echelon
  - -type of task (ops/log equip specific)

OUTPUT

- Collective Task Lists for Each Mission Phase
  - -Individual tasks
  - -Leader tasks
  - -Command Tasks
  - -Command & Satff tasks
- Training Info
  - -Related tasks
  - -Initial training time
  - -Next level coll task supported
  - -Rationale for selection for each task

TO 3.2 PRODUCE TASK ELEMENTS

FIGURE 11. Develop Individual Tasks, Conditions and Standards

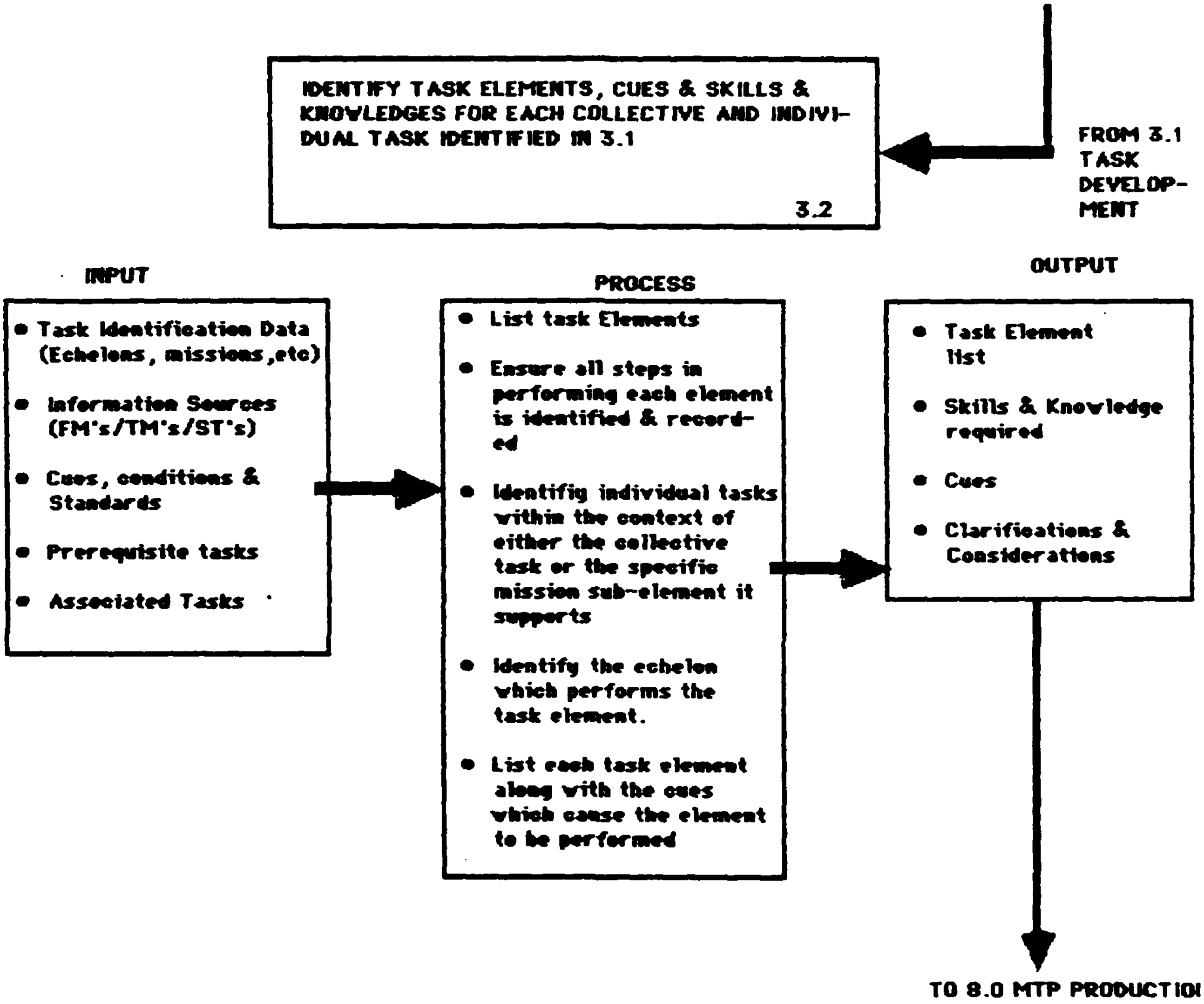

IDENTIFY TASK ELEMENTS, CUES & SKILLS & KNOWLEDGES FOR EACH COLLECTIVE AND INDIVIDUAL TASK IDENTIFIED IN 3.1
3.2
FROM 3.1 TASK DEVELOPMENT
INPUT
Task Identification Data (Echelons, missions,etc)
Information Sources (FM's/TM's/ST's)
Cues, conditions & Standards
Prerequisite tasks
Associated Tasks
PROCESS
List task Elements
Ensure all steps in performing each element is identified & recorded
Identifiy individual tasks within the context of either the collective task or the specific mission sub-element it supports
Identify the echelon which performs the task element.
List each task element along with the cues which cause the element to be performed
OUTPUT
Task Element list
Skills & Knowledge required
Cues
Clarifications & Considerations
TO 8.0 MTP PRODUCTIO

FIGURE 12. Develop Individual Tasks, Conditions and Standards

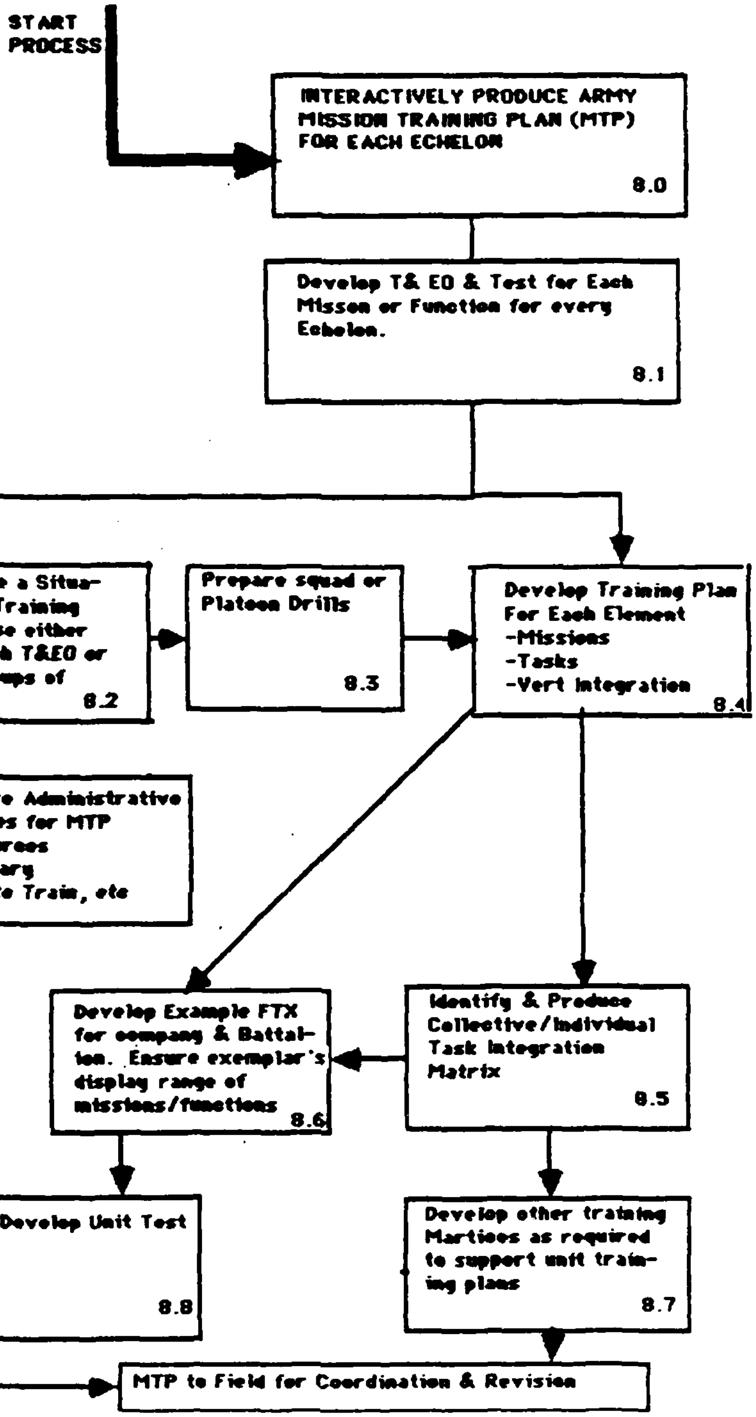
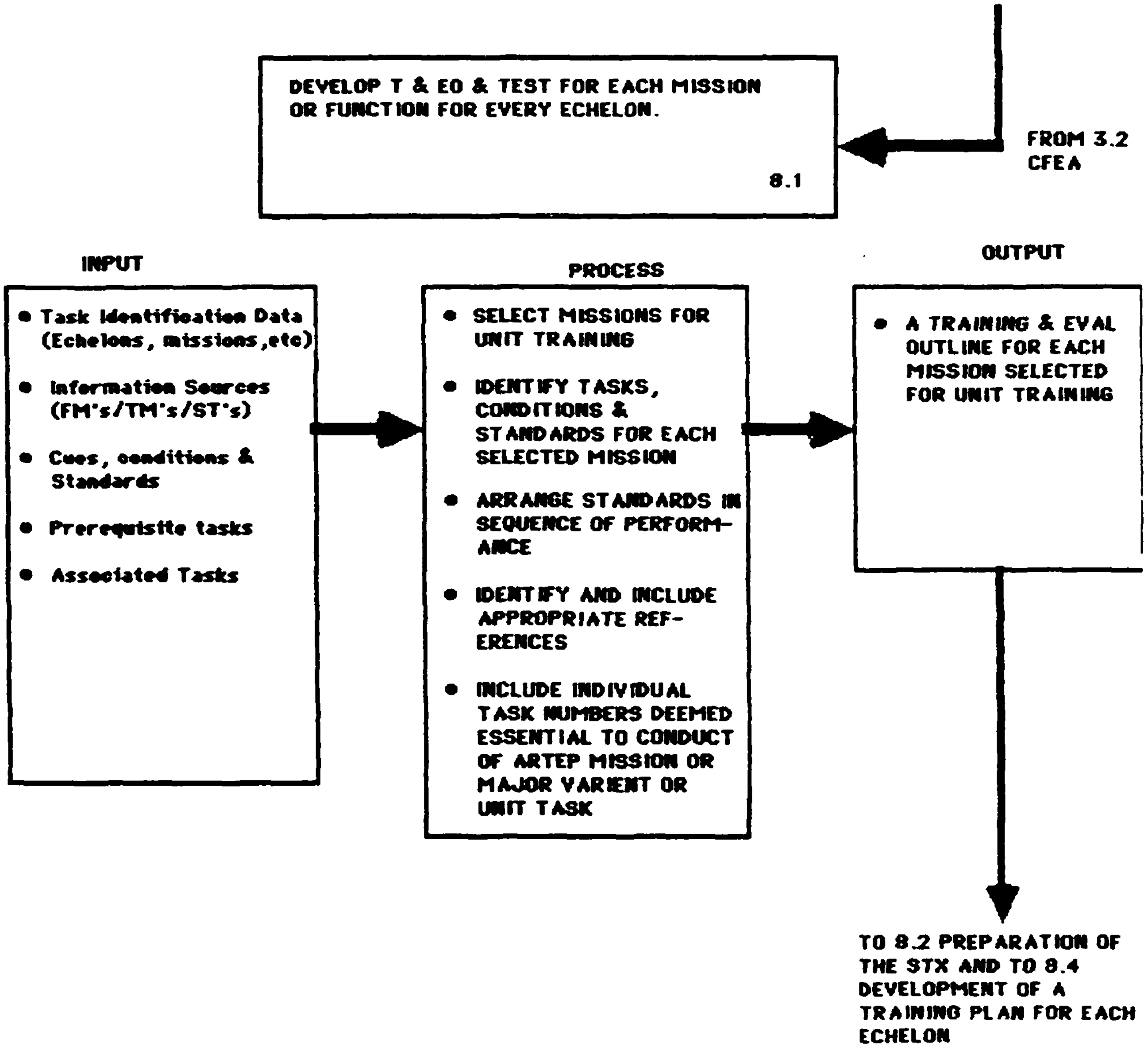

START PROCESS
INTERACTIVELY PRODUCE ARMY MISSION TRAINING PLAN (MTP) FOR EACH ECHELON 8.0
Develop T& EO & Test for Each Misson or Function for every Echelon. 8.1
Prepare a Situational Training Exercise either for each T&EO or for groups of T&EO 8.2
Prepare squad or Platoon Drills 8.3
Develop Training Plan For Each Element -Missions -Tasks -Vert Integration 8.4
Prepare Administrative Annexes for MTP -Resources -Glossary -How to Train, etc
Develop Example FTX for company & Battalion. Ensure exemplar's display range of missions/functions 8.6
Identify & Produce Collective/Individual Task Integration Matrix 8.5
Develop Unit Test 8.8
Develop other training Martices as required to support unit training plans 8.7
MTP to Field for Coordination & Revision

FIGURE 13. Identify Task Elements, Cues & Skills



DEVELOP T & EO & TEST FOR EACH MISSION OR FUNCTION FOR EVERY ECHELON.

8.1

FROM 3.2 CFEA

INPUT

- Task Identification Data (Echelons, missions, etc)
- Information Sources (FM's/TM's/ST's)
- Cues, conditions & Standards
- Prerequisite tasks
- Associated Tasks

PROCESS

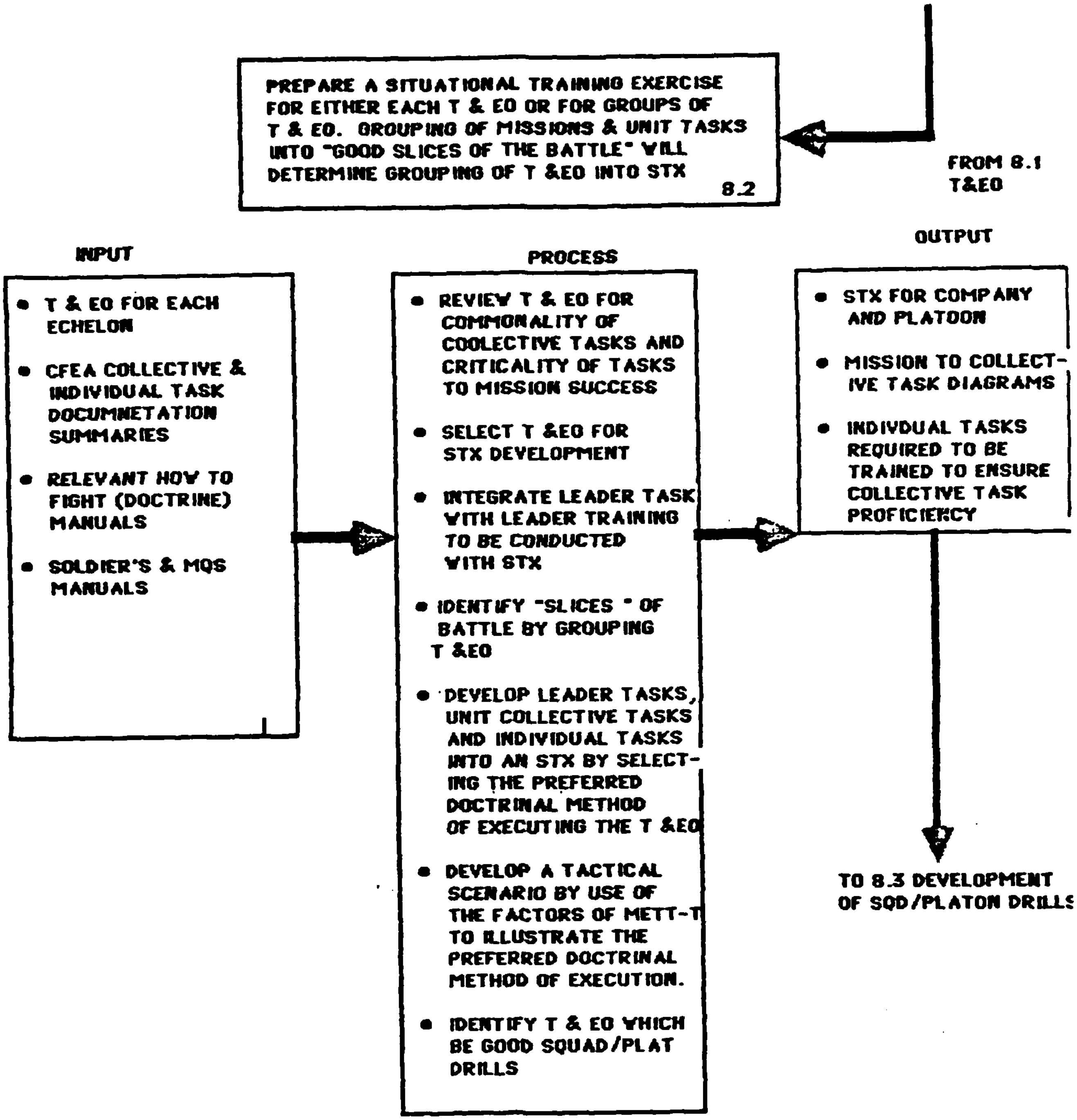
- SELECT MISSIONS FOR UNIT TRAINING
- IDENTIFY TASKS, CONDITIONS & STANDARDS FOR EACH SELECTED MISSION
- ARRANGE STANDARDS IN SEQUENCE OF PERFORMANCE
- IDENTIFY AND INCLUDE APPROPRIATE REFERENCES
- INCLUDE INDIVIDUAL TASK NUMBERS DEEMED ESSENTIAL TO CONDUCT OF ARTEP MISSION OR MAJOR VARIENT OR UNIT TASK

OUTPUT

- A TRAINING & EVAL OUTLINE FOR EACH MISSION SELECTED FOR UNIT TRAINING

TO 8.2 PREPARATION OF THE STX AND TO 8.4 DEVELOPMENT OF A TRAINING PLAN FOR EACH ECHELON

FIGURE 14. Develop T & EO



PREPARE A SITUATIONAL TRAINING EXERCISE FOR EITHER EACH T & EO OR FOR GROUPS OF T & EO. GROUPING OF MISSIONS & UNIT TASKS INTO "GOOD SLICES OF THE BATTLE" WILL DETERMINE GROUPING OF T &EO INTO STX 8.2

FROM 8.1 T&EO

INPUT

- T & EO FOR EACH ECHELON
- CFEA COLLECTIVE & INDIVIDUAL TASK DOCUMNETATION SUMMARIES
- RELEVANT HOW TO FIGHT (DOCTRINE) MANUALS
- SOLDIER'S & MQS MANUALS

PROCESS

- REVIEW T & EO FOR COMMONALITY OF COOLECTIVE TASKS AND CRITICALITY OF TASKS TO MISSION SUCCESS
- SELECT T &EO FOR STX DEVELOPMENT
- INTEGRATE LEADER TASK WITH LEADER TRAINING TO BE CONDUCTED WITH STX
- IDENTIFY "SLICES " OF BATTLE BY GROUPING T &EO
- DEVELOP LEADER TASKS, UNIT COLLECTIVE TASKS AND INDIVIDUAL TASKS INTO AN STX BY SELECTING THE PREFERRED DOCTRINAL METHOD OF EXECUTING THE T &EO
- DEVELOP A TACTICAL SCENARIO BY USE OF THE FACTORS OF METT-T TO ILLUSTRATE THE PREFERRED DOCTRINAL METHOD OF EXECUTION.
- IDENTIFY T & EO WHICH BE GOOD SQUAD/PLAT DRILLS

OUTPUT

- STX FOR COMPANY AND PLATOON
- MISSION TO COLLECTIVE TASK DIAGRAMS
- INDIVDUAL TASKS REQUIRED TO BE TRAINED TO ENSURE COLLECTIVE TASK PROFICIENCY

TO 8.3 DEVELOPMENT OF SQD/PLATON DRILLS

FIGURE 15. Prepare a Situational Training Exercise

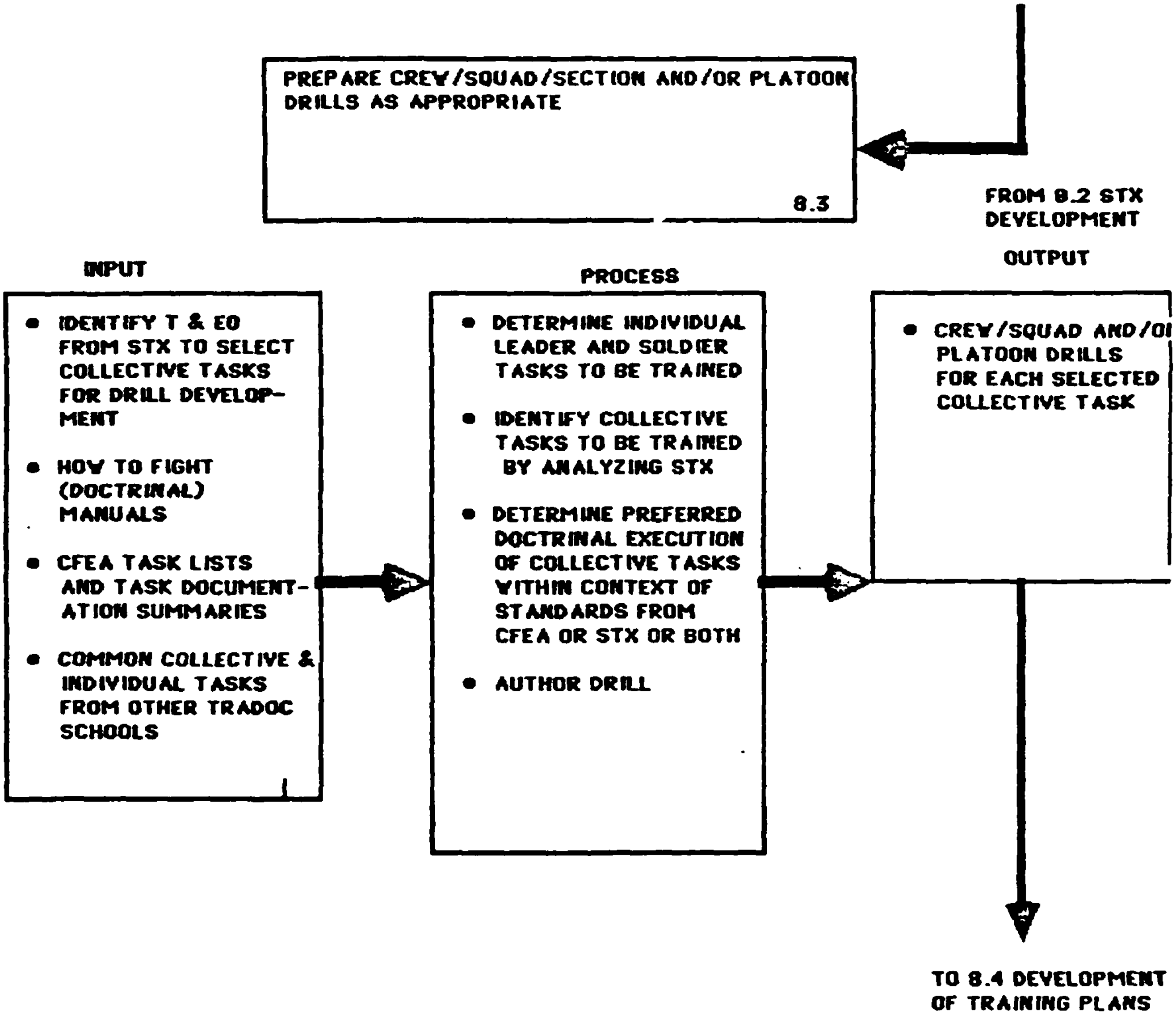
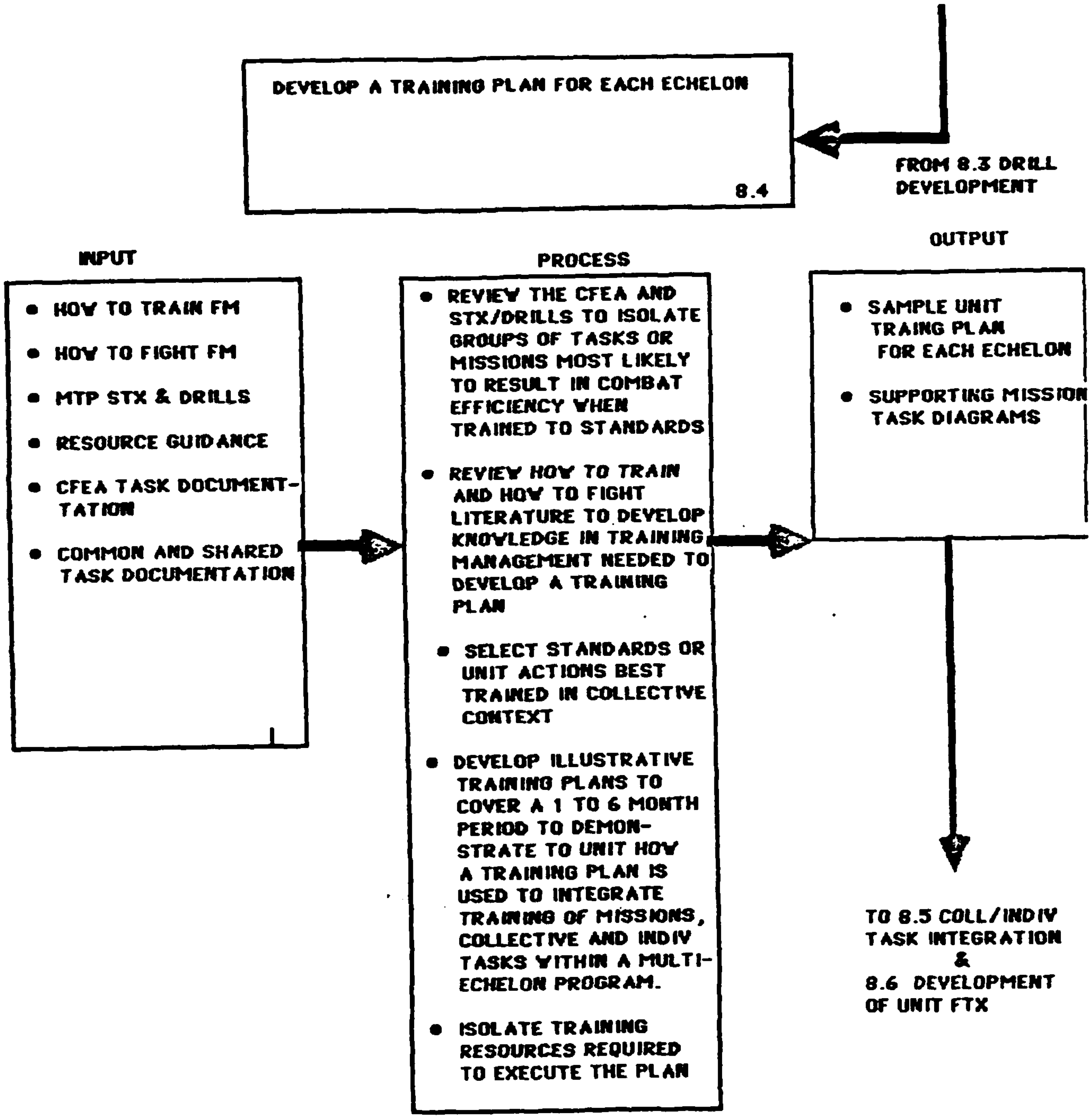

PREPARE CREW/SQUAD/SECTION AND/OR PLATOON
DRILLS AS APPROPRIATE
8.3
FROM 8.2 STX
DEVELOPMENT
INPUT
PROCESS
OUTPUT
• IDENTIFY T & EO FROM STX TO SELECT COLLECTIVE TASKS FOR DRILL DEVELOPMENT
• HOW TO FIGHT (DOCTRINAL) MANUALS
• CFEA TASK LISTS AND TASK DOCUMENTATION SUMMARIES
• COMMON COLLECTIVE & INDIVIDUAL TASKS FROM OTHER TRADOC SCHOOLS
• DETERMINE INDIVIDUAL LEADER AND SOLDIER TASKS TO BE TRAINED
• IDENTIFY COLLECTIVE TASKS TO BE TRAINED BY ANALYZING STX
• DETERMINE PREFERRED DOCTRINAL EXECUTION OF COLLECTIVE TASKS WITHIN CONTEXT OF STANDARDS FROM CFEA OR STX OR BOTH
• AUTHOR DRILL
• CREW/SQUAD AND/OI PLATOON DRILLS FOR EACH SELECTED COLLECTIVE TASK
TO 8.4 DEVELOPMENT
OF TRAINING PLANS

FIGURE 16. Prepare Crew/Squad/Section and/or Platoon Drills



DEVELOP A TRAINING PLAN FOR EACH ECHELON

8.4

FROM 8.3 DRILL DEVELOPMENT

INPUT

- HOW TO TRAIN FM
- HOW TO FIGHT FM
- MTP STX & DRILLS
- RESOURCE GUIDANCE
- CFEA TASK DOCUMENT-TATION
- COMMON AND SHARED TASK DOCUMENTATION

PROCESS

- REVIEW THE CFEA AND STX/DRILLS TO ISOLATE GROUPS OF TASKS OR MISSIONS MOST LIKELY TO RESULT IN COMBAT EFFICIENCY WHEN TRAINED TO STANDARDS
- REVIEW *HOW TO TRAIN* AND HOW TO FIGHT LITERATURE TO DEVELOP KNOWLEDGE IN TRAINING MANAGEMENT NEEDED TO DEVELOP A TRAINING PLAN
- SELECT STANDARDS OR UNIT ACTIONS BEST TRAINED IN COLLECTIVE CONTEXT
- DEVELOP ILLUSTRATIVE TRAINING PLANS TO COVER A 1 TO 6 MONTH PERIOD TO DEMONSTRATE TO UNIT HOW A TRAINING PLAN IS USED TO INTEGRATE TRAINING OF MISSIONS, COLLECTIVE AND INDIV TASKS WITHIN A MULTI-ECHELON PROGRAM.
- ISOLATE TRAINING RESOURCES REQUIRED TO EXECUTE THE PLAN

OUTPUT

- SAMPLE UNIT TRAING PLAN FOR EACH ECHELON
- SUPPORTING MISSION TASK DIAGRAMS

TO 8.5 COLL/INDIV TASK INTEGRATION
&
8.6 DEVELOPMENT OF UNIT FTX

FIGURE 17. Develop a Training Plan

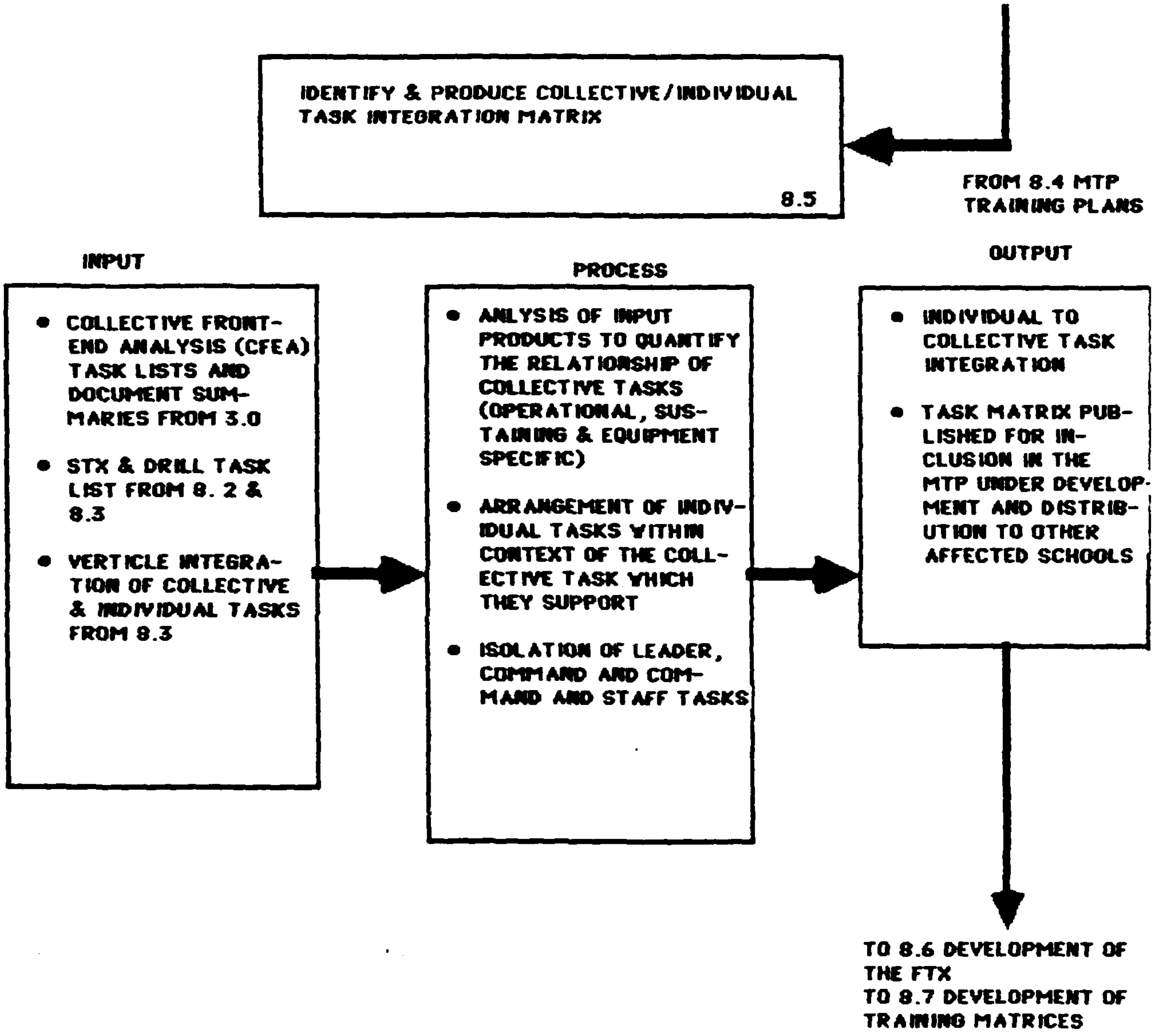
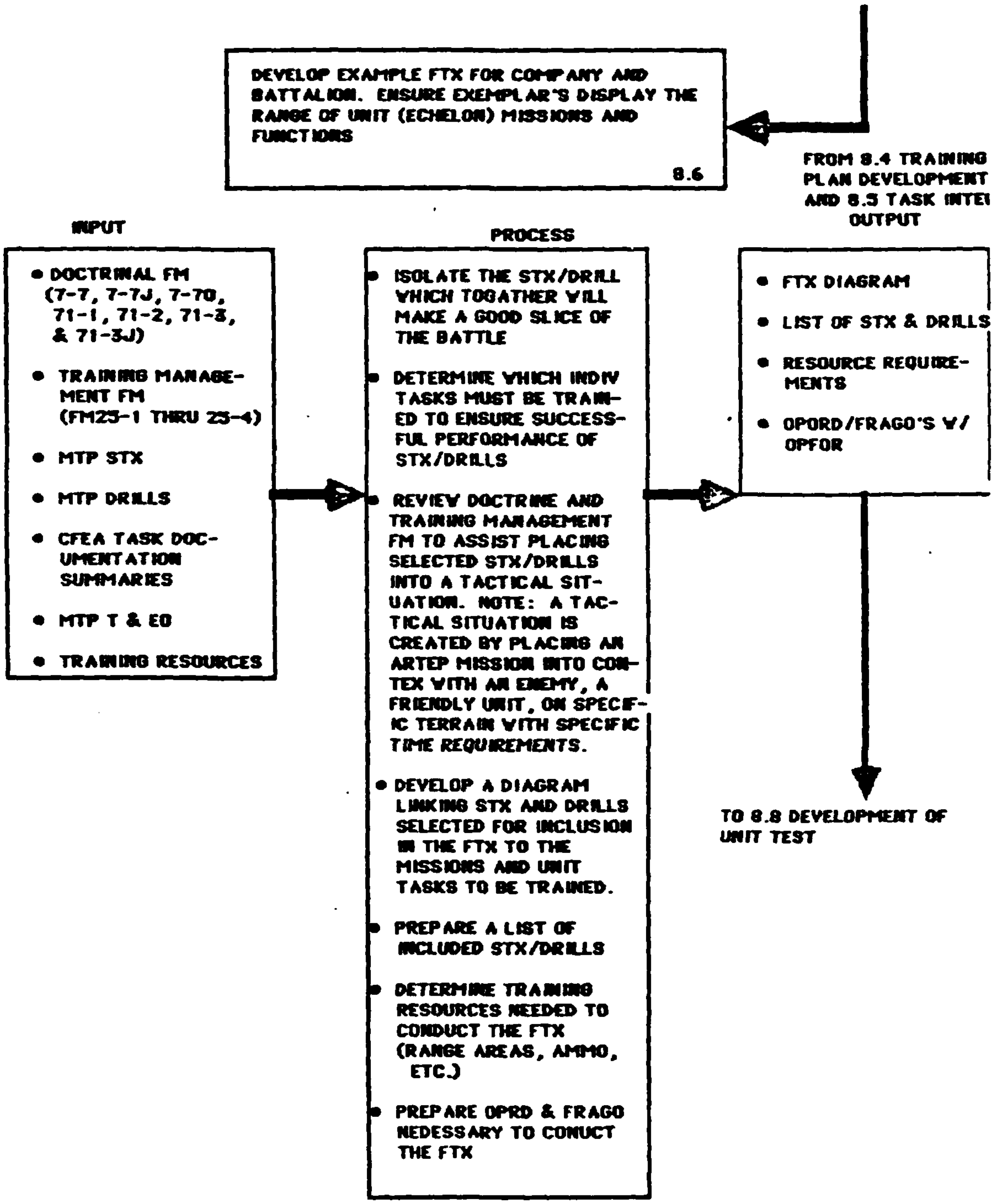

IDENTIFY & PRODUCE COLLECTIVE/INDIVIDUAL TASK INTEGRATION MATRIX
8.5
FROM 8.4 MTP TRAINING PLANS
INPUT
PROCESS
OUTPUT
• COLLECTIVE FRONT-END ANALYSIS (CFEA) TASK LISTS AND DOCUMENT SUMMARIES FROM 3.0
• STX & DRILL TASK LIST FROM 8. 2 & 8.3
• VERTICLE INTEGRATION OF COLLECTIVE & INDIVIDUAL TASKS FROM 8.3
• ANLYSIS OF INPUT PRODUCTS TO QUANTIFY THE RELATIONSHIP OF COLLECTIVE TASKS (OPERATIONAL, SUSTAINING & EQUIPMENT SPECIFIC)
• ARRANGEMENT OF INDIVIDUAL TASKS YITHIN CONTEXT OF THE COLLECTIVE TASK YHICH THEY SUPPORT
• ISOLATION OF LEADER, COMMAND AND COMMAND AND STAFF TASKS
• INDIVIDUAL TO COLLECTIVE TASK INTEGRATION
• TASK MATRIX PUBLISHED FOR INCLUSION IN THE MTP UNDER DEVELOPMENT AND DISTRIBUTION TO OTHER AFFECTED SCHOOLS
TO 8.6 DEVELOPMENT OF THE FTX
TO 8.7 DEVELOPMENT OF TRAINING MATRICES

FIGURE 18. Identify and Produce Task Integration Matrix



DEVELOP EXAMPLE FTX FOR COMPANY AND BATTALION. ENSURE EXEMPLAR'S DISPLAY THE RANGE OF UNIT (ECHELON) MISSIONS AND FUNCTIONS

8.6

FROM 8.4 TRAINING PLAN DEVELOPMENT AND 8.5 TASK INTE

INPUT

- DOCTRINAL FM (7-7, 7-7J, 7-70, 71-1, 71-2, 71-3, & 71-3J)
- TRAINING MANAGEMENT FM (FM25-1 THRU 25-4)
- MTP STX
- MTP DRILLS
- CFEA TASK DOCUMENTATION SUMMARIES
- MTP T & EO
- TRAINING RESOURCES

PROCESS

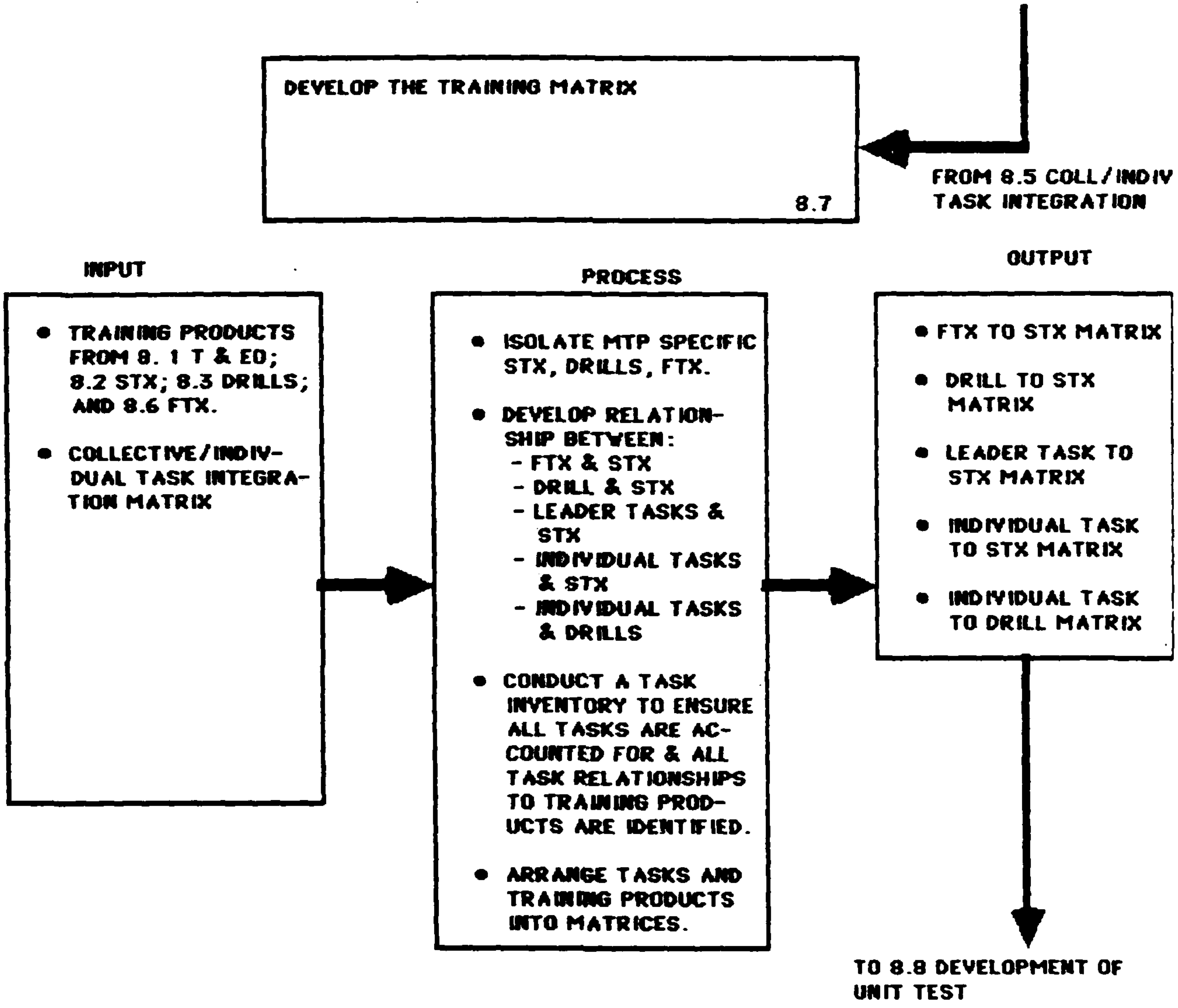
- ISOLATE THE STX/DRILL WHICH TOGATHER VILL MAKE A GOOD SLICE OF THE BATTLE
- DETERMINE WHICH INDIV TASKS MUST BE TRAINED TO ENSURE SUCCESSFUL PERFORMANCE OF STX/DRILLS
- REVIEW DOCTRINE AND TRAINING MANAGEMENT FM TO ASSIST PLACING SELECTED STX/DRILLS INTO A TACTICAL SITUATION. NOTE: A TACTICAL SITUATION IS CREATED BY PLACING AN ARTEP MISSION INTO CONTEX WITH AN ENEMY, A FRIENDLY UNIT, ON SPECIFIC TERRAIN WITH SPECIFIC TIME REQUIREMENTS.
- DEVELOP A DIAGRAM LINKING STX AND DRILLS SELECTED FOR INCLUSION IN THE FTX TO THE MISSIONS AND UNIT TASKS TO BE TRAINED.
- PREPARE A LIST OF INCLUDED STX/DRILLS
- DETERMINE TRAINING RESOURCES NEEDED TO CONDUCT THE FTX (RANGE AREAS, AMMO, ETC.)
- PREPARE OPRD & FRAGO NEDESSARY TO CONUCT THE FTX

OUTPUT

- FTX DIAGRAM
- LIST OF STX & DRILLS
- RESOURCE REQUIREMENTS
- OPORD/FRAGO'S W/ OPFOR

TO 8.8 DEVELOPMENT OF UNIT TEST

FIGURE 19. Develop FTX



DEVELOP THE TRAINING MATRIX

8.7

FROM 8.5 COLL/INDIV TASK INTEGRATION

INPUT

- TRAINING PRODUCTS FROM 8. 1 T & EO; 8.2 STX; 8.3 DRILLS; AND 8.6 FTX.
- COLLECTIVE/INDIVIDUAL TASK INTEGRATION MATRIX

PROCESS

- ISOLATE MTP SPECIFIC STX, DRILLS, FTX.
- DEVELOP RELATIONSHIP BETWEEN:
  - FTX & STX
  - DRILL & STX
  - LEADER TASKS & STX
  - INDIVIDUAL TASKS & STX
  - INDIVIDUAL TASKS & DRILLS
- CONDUCT A TASK INVENTORY TO ENSURE ALL TASKS ARE ACCOUNTED FOR & ALL TASK RELATIONSHIPS TO TRAINING PRODUCTS ARE IDENTIFIED.
- ARRANGE TASKS AND TRAINING PRODUCTS INTO MATRICES.

OUTPUT

- FTX TO STX MATRIX
- DRILL TO STX MATRIX
- LEADER TASK TO STX MATRIX
- INDIVIDUAL TASK TO STX MATRIX
- INDIVIDUAL TASK TO DRILL MATRIX

TO 8.8 DEVELOPMENT OF UNIT TEST

FIGURE 20. Develop the Training Matrix

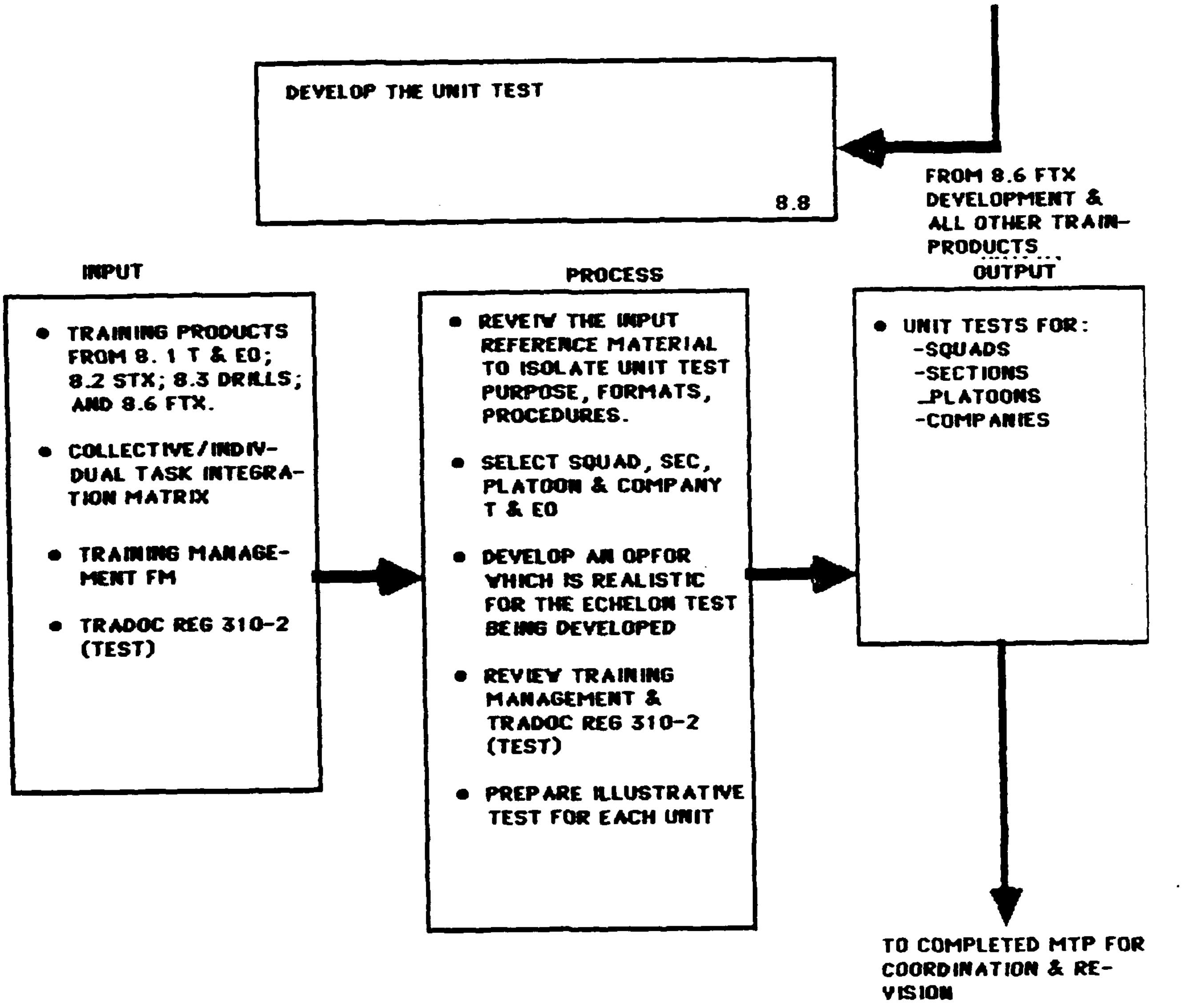

DEVELOP THE UNIT TEST
8.8
FROM 8.6 FTX DEVELOPMENT & ALL OTHER TRAIN-PRODUCTS
INPUT
• TRAINING PRODUCTS FROM 8. 1 T & EO; 8.2 STX; 8.3 DRILLS; AND 8.6 FTX.
• COLLECTIVE/INDIVDUAL TASK INTEGRATION MATRIX
• TRAINING MANAGEMENT FM
• TRADOC REG 310-2 (TEST)
PROCESS
• REVEIW THE INPUT REFERENCE MATERIAL TO ISOLATE UNIT TEST PURPOSE, FORMATS, PROCEDURES.
• SELECT SQUAD, SEC, PLATOON & COMPANY T & EO
• DEVELOP AN OPFOR WHICH IS REALISTIC FOR THE ECHELON TEST BEING DEVELOPED
• REVIEW TRAINING MANAGEMENT & TRADOC REG 310-2 (TEST)
• PREPARE ILLUSTRATIVE TEST FOR EACH UNIT
OUTPUT
• UNIT TESTS FOR:
-SQUADS
-SECTIONS
_PLATOONS
-COMPANIES
TO COMPLETED MTP FOR COORDINATION & REVISION

FIGURE 21. Develop the Unit Test

3. Time delay tolerance. What is the time delay allowed between receiving the task cue and starting the task performance?

4. Consequence of inadequate performance. What other tasks are affected?

5. Immediacy of performance. How soon after achieving task proficiency can the crew/platoon/individual be expected to perform the task again?

6. Task importance. How important is an individual task to collective/unit success. Another way to ask this question is to determine how often a collective task is a component of an ARTEP mission.

7. Combat criticality. Can the task only be performed in combat, or can it also be performed in administrative or training situations? An example can be found in the tasks of decontamination of nerve agents and TOW missile firing. In both cases the tasks are combat critical. Decontamination of actual nerve gas agents can be accomplished in combat only since the US Army is not allowed to train with live nerve agents. TOW missiles however, are combat critical but can be trained to a limited degree only (due to the cost and consequent limited availability of TOW missiles) in time of peace.

8. Proficiency decay rate. How often must a task be performed to ensure that skills are not reduced below performance measure standards considering time, personnel turbulence and task difficulty?

Obviously the answers to these questions are dependent on unit conditions and the amount and quality of data available to the MTP author. However, answers are necessary to develop an effective MTP. CAPS developers should examine the database elements to determine the best design for a system to assist the author in answering them.

Once the MTP author makes decisions as to task criticality, he faces ARTEP-unique problems of data aggregation that must be solved if he is to produce an effective Mission Training Plan (MTP). Remember, the MTP as now designed describes unit training requirements in a way that reflects the results of a training oriented analysis of collective tasks.

The MTP author must identify slices of the battle that could be practiced as specific drills or Situational Training Exercises (STXs) (see Appendix E, Glossary). The author must also develop "Training Roadmaps," which link skills and tasks to missions in a logical manner so that attempts to train a difficult task are not made until all included, less difficult tasks have been mastered. In a training situation that mandates multi-echelon, integrated training, this "roadmapping" requirement is an obvious candidate for automated support.

Last but not least, the MTP author must deal with the integration of individual and collective tasks as well as the integration of collective tasks to ARTEP missions and all the other matrices required by TRADOC Reg 310-2 (Test). (i.e., FTX-to-STX, Drill-to-Mission, Leader Task-to-STX, Individual Task-to-STX). Here again, automated support could vastly simplify his job.

In sum, the MTP author must conduct a training collective task analysis, using the products of a doctrinal collective task analysis. This process is illustrated by the diagrams in Figure 4 above, that show the CFEA data flow from the database to the Training & Evaluation Outline (T&EO) thru the MTP authors of DOTD and CATD. With the development of the T&EO, the MTP authors start the process of describing unit training requirements, a process that ends with the completed Mission Training Plan (MTP). Field Training Exercises (FTXs) are developed as part of the Mission Training Plan, to illustrate to commanders and unit trainers where the drills and STXs come together as a unit tactical capability within the context of unit missions.

Figure 22 depicts twelve sequentially interrelated processes, each of which involves a number of internal steps that are necessary to produce an MTP. CAPS will be designed to support these processes, beginning with the development of detailed T&EO and tests ending with the ARTEP Mission Training Plan Development. Coordination will occur "off-line." While revisions will input to the CAPS system, this input will occur at the CFEA module as any other CFEA input. Printing and preparation by the Department of the Army is outside the CAPS design at this time.

Preparing Training and Evaluation Outlines (T&EO). A detailed T&EO consists of (1) one or more missions or function(s) of a specific unit (e.g., battalion, squad; by type), (2) conditions, (3) subtasks, (4) standards arranged in the proper sequence of performance, (5) references, and (6) task numbers of separate individual tasks deemed essential to the successful conduct of the ARTEP mission or a subtask.

The T&EO is developed by the ARTEP author, based on the data provided by the CFEA. The CFEA identifies all unit missions for each echelon from squad to battalion. The School commandant convenes a selection board to determine missions that are essential for all like units to perform. Selected missions are incorporated into the MTP by the preparation of a T&EO and become the core for which all collective tasks, drills, leader tasks and STX are to be developed. T&EOs, which support other missions that are deemed less essential or are unique to specific units, will also be included after all essential missions have been developed.

Detailed information regarding preparation of a T&EO is contained in Chapter 3 of TRADOC Regulation 310-2. A sample T&EO is shown in Table 4.

Preparing the Situational Training Exercise (STX). STX are mission-oriented exercises, designed to be of short duration, and are simple field exercises to train a group of closely related collective tasks. They are illustrative in nature as they depict a doctrinally preferred method of executing the collective tasks to established standards. The STX is developed after the completion of the T&EOs, which have been prepared for each essential mission.

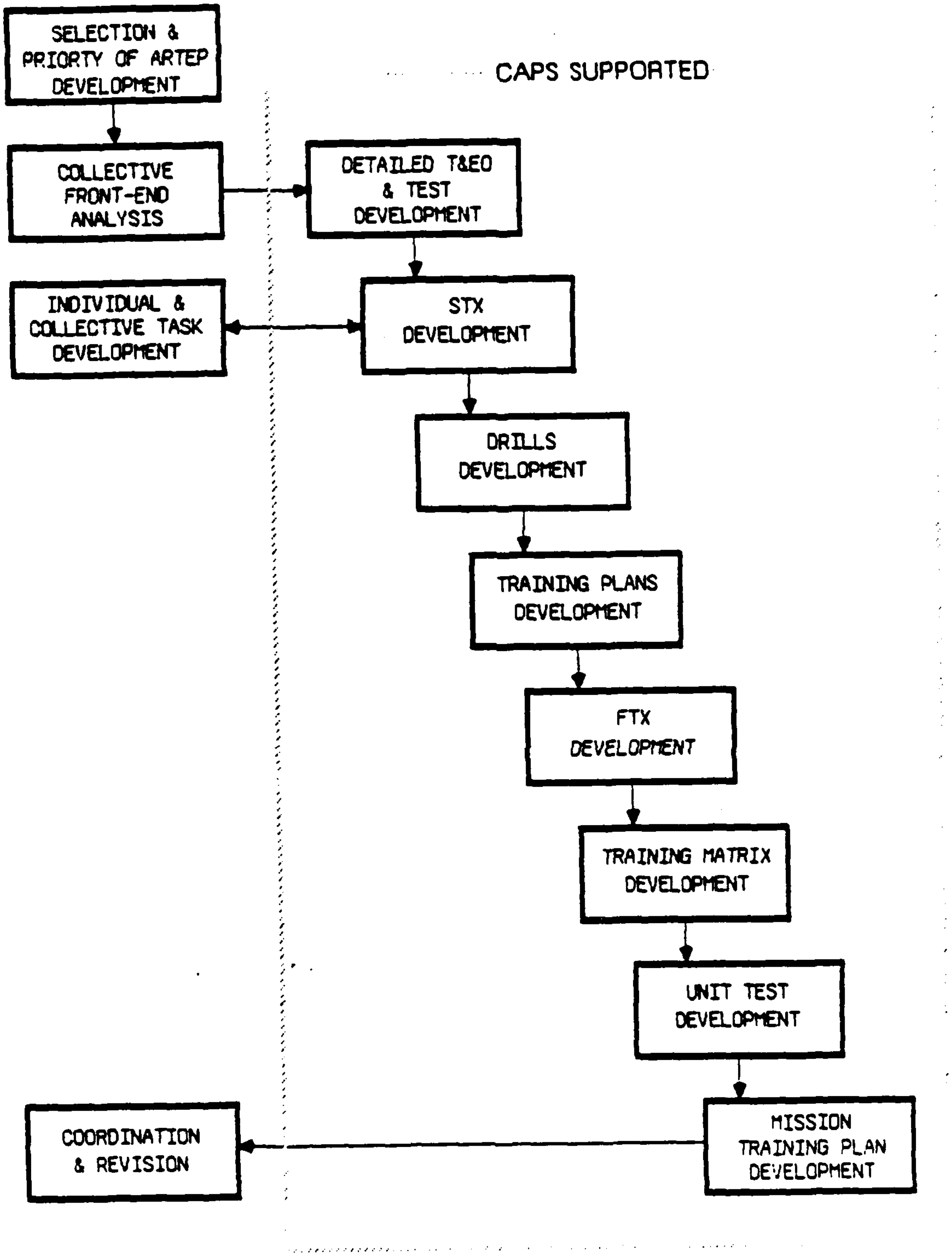

SELECTION & PRIORTY OF ARTEP DEVELOPMENT
CAPS SUPPORTED
COLLECTIVE FRONT-END ANALYSIS
DETAILED T&EO & TEST DEVELOPMENT
INDIVIDUAL & COLLECTIVE TASK DEVELOPMENT
STX DEVELOPMENT
DRILLS DEVELOPMENT
TRAINING PLANS DEVELOPMENT
FTX DEVELOPMENT
TRAINING MATRIX DEVELOPMENT
UNIT TEST DEVELOPMENT
COORDINATION & REVISION
MISSION TRAINING PLAN DEVELOPMENT

FIGURE 22. Processes in the Development of the MTP

TABLE 4

SAMPLE TRAINING & EVALUATION OUTLINE
(From Training Circular 17-15-1,
Tank Platoon Mission Training Plan, dated Jun '84)

TASK NUMBER: 076 (Task Number is assigned during CFEA)

TASK STATEMENT: Execute a Hasty Attack

CONDITIONS: Day or night, under any climatic conditions as part of offensive operations. The OPFOR may be static or moving.

STANDARDS:

Subtasks: yes no NE (not eval)

The Platoon:

1. Reacts to OPFOR presence without hesitation

2. Initiates fire and morement to destroy OPFOR before being fixed by the OPFOR.

3. Conducts the assault.

   a. Insuring that maneuver elements are overwatched.

   b. Using suppressive fire against the OPFOR.

   c. So that maximum available combat power is concentrated against the OPFOR at the decisive time and place as determined by commander.

References: FM 17-15 (test); Division 86 Tank Plt., Chpt 3, Para 3.2 & 3.3
FM 17-19E 1,2, & 3; Soldier's Manual for MOS 19E

Individual Task Numbers:
Platoon Leader: 171-140-4016
171-326-3049
Soldier's Manual
Skill Level 3: 071-331-0820
171-127-1001
etc.
Skill Level 1&2: 071-326-0608
113-571-1005
etc.

NOTE: This example differs from TRADOC Regulation 310-2 in that the evaluation columns grade on Yes/No or not evaluated. The regulation calls for a five column evaluation format ranging from 1-5.

Components of the STX are designed to integrate leader training, leader tasks, drills, and individual soldier tasks into a realistic combat scenario. Trainers may execute them as written or tailor them to meet specific training objectives. In either case, the performance measures are to be executed to prescribed standards.

Application of the STX, due to the design criteria above, are applicable primarily at the platoon and company level. If, however, the MTP author determines that higher level command and staff tasks require STX to train specific tasks or drills most efficiently, the appropriate STX should be developed and integrated into the MTP.

STX combine the "How-to-Train" specifics for selected collective tasks with the "What to Train" found in doctrinal manuals. STX have define start and stop points and are used repetitively to train to unit criteria.

The following questions must be answered by the MTP author when he designs the STX:

1. How critical are the missions and unit functions to the combat capability of the unit. Such factors as stated missions, critical combat collective tasks and implied missions, all available in the CFEA, help the author narrow the focus of STX development.

2. What slices of the battle make good training exercises? The author must make a training analysis of the missions and tasks identified in the CFEA to aggregate the data into reasonable chunks of time, personnel and tasks. An example would be the ARTEP mission for a platoon of the Hasty Attack.

Let us decompose this mission and see the elements of decision making and design that goes into the authoring of an STX. We will follow this with the presentation of a sample Hasty Attack STX. To begin, the ARTEP author reviews the task mission and task list from the CFEA. Under "Missions" he finds the platoon and squad execute the hasty attack as part of the movement to contact. He finds further that execution requires platoon proficiency in unit functional areas of "Fire and Movement," "Movement" "Shoot, "Sustain" and "NBC." This tells him that an STX designed to train a platoon in the hasty attack must contain drills, individual tasks, and leader tasks from all of these functional areas. Because the database contains the vertical and horizontal integration of tasks, missions, and echelons (individual to platoon in this case), the ARTEP author can now list and diagram all collective tasks in the hasty attack ARTEP mission. We see this below in Figure 23.

Once the MTTP author has diagrammed the proposed STX, he goes back to the CFEA to determine which individual and leader tasks are required to allow the platoon to execute each of the collective sub-unit tasks. He will list the tasks for use in the published STX/MTP. He will also, at this time be looking for ARTEP tasks within the STX diagram (as shown in Figure 23) that will be good candidates for development as drills. We will discuss development of

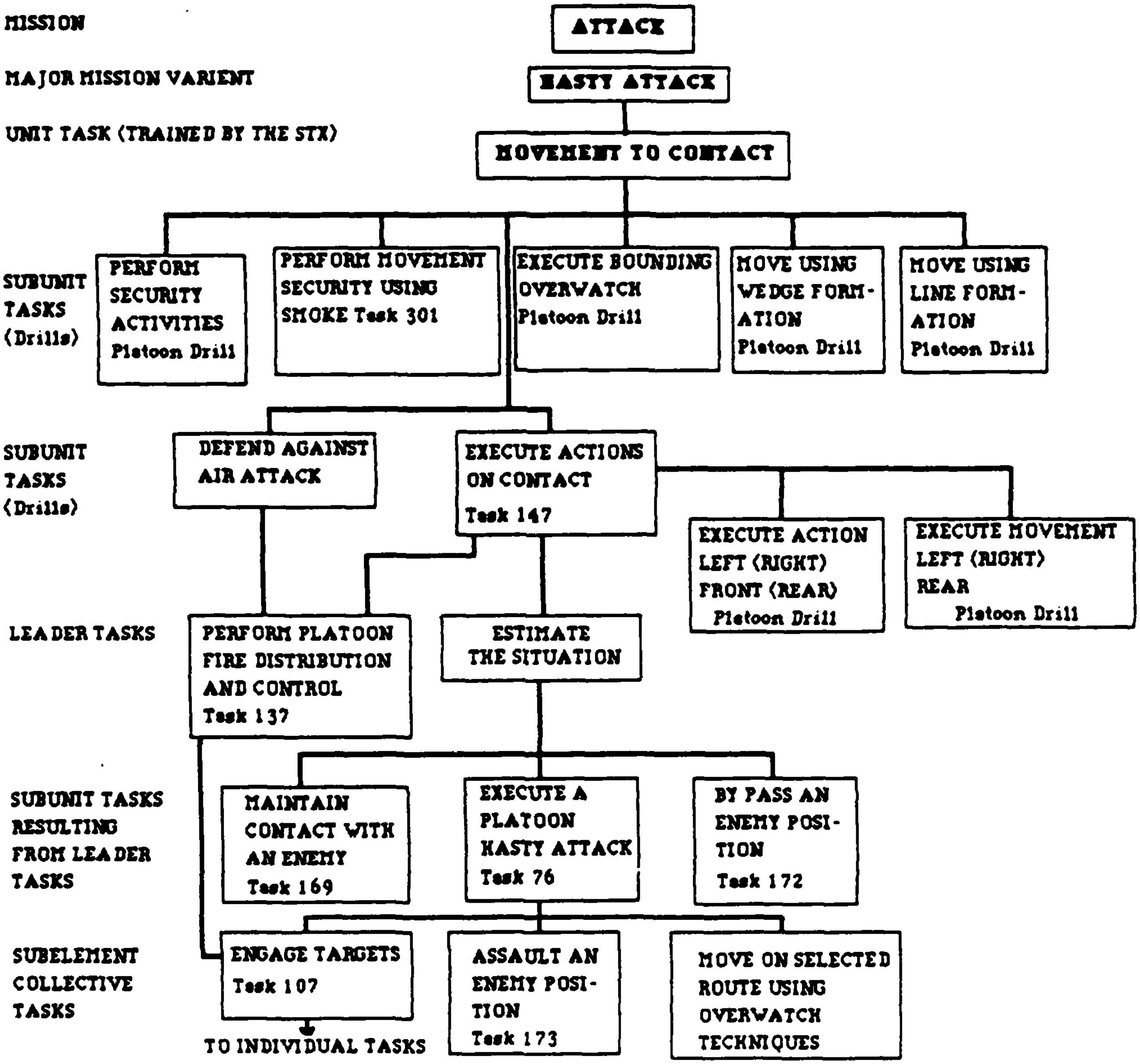

MISSION
MAJOR MISSION VARIENT
UNIT TASK (TRAINED BY THE STX)
SUBUNIT TASKS (Drills)
SUBUNIT TASKS (Drills)
LEADER TASKS
SUBUNIT TASKS RESULTING FROM LEADER TASKS
SUBELEMENT COLLECTIVE TASKS
ATTACK
HASTY ATTACK
MOVEMENT TO CONTACT
PERFORM SECURITY ACTIVITIES Platoon Drill
PERFORM MOVEMENT SECURITY USING SMOKE Task 301
EXECUTE BOUNDING OVERWATCH Platoon Drill
MOVE USING WEDGE FORMATION Platoon Drill
MOVE USING LINE FORMATION Platoon Drill
DEFEND AGAINST AIR ATTACK
EXECUTE ACTIONS ON CONTACT Task 147
EXECUTE ACTION LEFT (RIGHT) FRONT (REAR) Platoon Drill
EXECUTE MOVEMENT LEFT (RIGHT) REAR Platoon Drill
PERFORM PLATOON FIRE DISTRIBUTION AND CONTROL Task 137
ESTIMATE THE SITUATION
MAINTAIN CONTACT WITH AN ENEMY Task 169
EXECUTE A PLATOON HASTY ATTACK Task 76
BY PASS AN ENEMY POSITION Task 172
ENGAGE TARGETS Task 107
ASSAULT AN ENEMY POSITION Task 173
MOVE ON SELECTED ROUTE USING OVERWATCH TECHNIQUES
TO INDIVIDUAL TASKS

FIGURE 23. Example STX: Platoon Hasty Attack

drills in detail in the next paragraph. At this time it is sufficient for the reader to know that it is here that the best look is made to identify potential drills. The next step in STX development is to turn to FM 25-3, How to Train, and develop the "Tips to Trainers" portion of the STX. Using this Field Manual, and his experience, the author discusses the leader's steps in preparing for the conduct of the STX. The remainder of the STX consists of the preparation of the Fragmentary Order (FRAGO) that establishes the "Situation" in the Situational Training Exercise. A FRAGO looks like this:

> "(Your Platoon), IN CONTACT WITH SQUAD-SIZED ENEMY FORCE. (Your Platoon), SIEZE OBJECTIVE __________, COORDINATES (__________).
> __________ PLATOON (Simulated) and (Sister platoon) BASE OF FIRE COORDINATES (__________). ACKNOWLEDGE.

The FRAGO above provides all that the Platoon Leader/Sgt needs to know to plan an exercise. To Wit: The Mission - "Seize an Objective", the Enemy - "a squad-size force", Troops - "Your platoon supported by two other platoons", Terrain - as specified by "Coordinates", and the Time - implied "As soon as possible." Now, the ARTEP author has only to establish the Standards for the STX and he has a completed product. The Standards come directly out of the CFEA data base, Phase V, product 3. A standard is prepared/published for each task contained in the STX. After development of the drills, which will be integrated into this STX, the drill standards will be used for each task that is trained by drill performance.

Remember, that the MTP author must also develop "roadmaps," which link skills and tasks to ARTEP missions. Now lets look at the Platoon "Roadmap" as this is really the Mission Training Plan "Big Picture." From Figure 24 we can see the "roadmap" emerging. Look at the top of the triangle. The platoon ARTEP missions, derived from the CFEA are all listed, including our example "Hasty Attack." Put any Mission at the top and the MTP author can go down thru the collective tasks of command and control and supplemental tasks such as Control Combat Operations, reporting, and maintaining operational security and get right into the individual leader and individual soldier tasks. From this "Training Analysis," the MTP author can develop his collective-to-individual task integration matrices, as shown and discussed previously in Figure 7. He is then ready to develop the squad/platoon drills.

Development of Drills. Drills are used to train small units (infantry squad or platoon, tank crew or platoon) to a peak of performance on those collective tasks that are always performed in the same way. In other words, the purpose of the drill literally is to condition each member of a unit to perform his individual tasks within the context of the collective task, the same way time after time. An analogy can be made to the football squad. The actions of each team member are practiced for each play over and over so that such actions as pass-run coordination, or the bump and run, are executed to precise standards of timing and individual skills. So it is with a tank crew engaging a tank target or a Bradley Fighting Vehicle Squad dismounting for a dismounted assault.

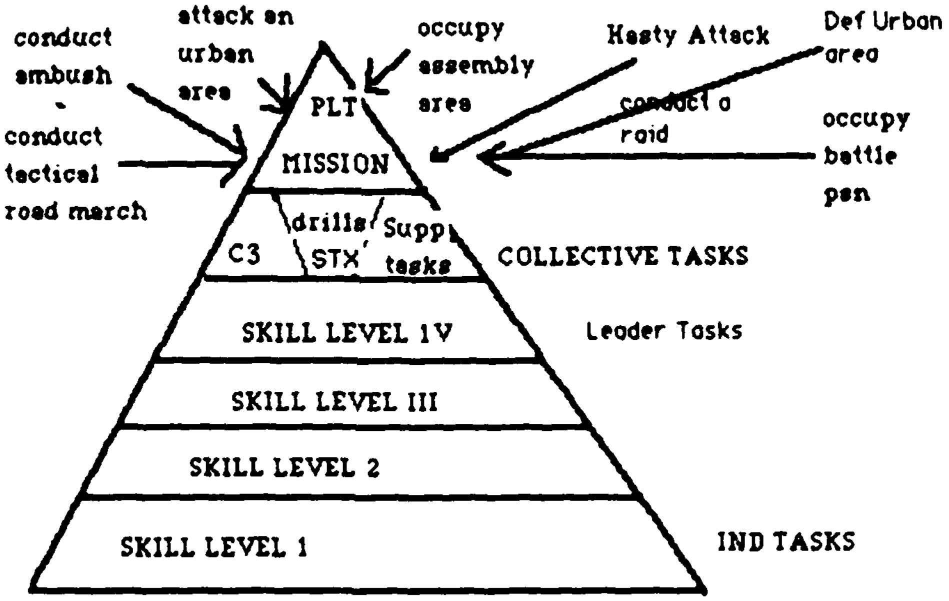


FIGURE 24. Platoon Mission Training Plan

Drills then describe a small portion of the battlefield and are to an STX as a play is to a series of plays. They are designed to capitalize on a specific situation. Drills are comprised of both collective and individual tasks, designed to hone collective skills. To develop a drill, the MTP author looks at his inventory of STX and by analysis determines which collective task he selected as a drill candidate during STX development. Using the same "Hasty Attack" STX we examined in the discussion on STX development and Figure 23, we can see that the following tasks are good candidates for drill development as the prime training vehicle:

1. All of the platoon movements that occur in fixed formations such as (1) execute bounding overwatch, (2) move using a wedge formation, (3) a line formation, and (4) execute action right and left, are good candidates as they are always done in the same manner and are initiated on the same cue, either enemy action or visual signal.

2. Defense against air attack and surveillance activities are also candidates for the same reasons. Each requires a uniform collective response and each has the stability of uniform individual tasks.

Once again we see the relevance of asking the MTP author to do a training analysis, which will identify slices of the battle and integrate individual and collective tasks with ARTEP missions. Do these tasks require collective practice? Clearly, the answer is "yes." Can we identify tasks that apply to leaders? Again, the answer is "yes." Leaders designate the routes and formations and determine the direction of the platoon's actions. Unit functions related to movement and fire and maneuver are involved and Soldier's Manual tasks can be clearly identified and integrated into the drill (map reading, driving, visual signals, etc.).

Let us look at the Hasty Attack Diagram after the author has conducted his analysis of those tasks that should be supported by drills. An illustration is shown at Figure 25.

To develop an MTP drill from this point the author proceeds as follows:

1. First, he determines the individual leader and soldier tasks by task number, title, reference, and standard for each selected drill within the STX.
2. Secondly, he selects/identifies each collective task by task number, title, and reference for the squad/crew and for the platoon.
3. When he has assembled these data, he starts to author the drill.

In this case the STX we are working on is the "Hasty Attack" and we are going to develop the "Move by Wedge Formation" Platoon Drill. Drill-to-ARTEP Task linkage is a list of ARTEP missions, by title and number that will be supported by this drill. They are:

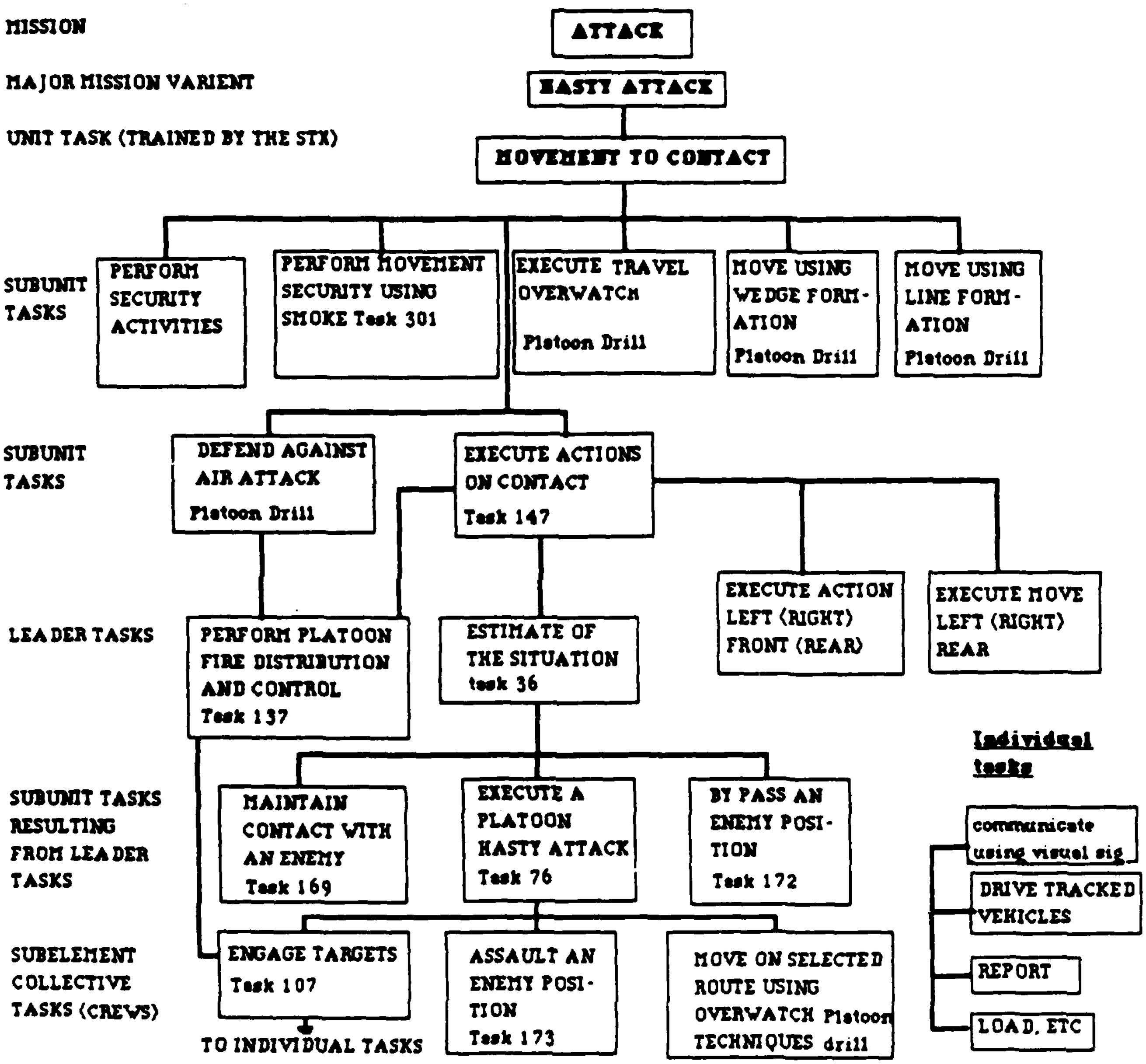

MISSION
MAJOR MISSION VARIENT
UNIT TASK (TRAINED BY THE STX)
SUBUNIT TASKS
SUBUNIT TASKS
LEADER TASKS
SUBUNIT TASKS RESULTING FROM LEADER TASKS
SUBELEMENT COLLECTIVE TASKS (CREWS)
ATTACK
HASTY ATTACK
MOVEMENT TO CONTACT
PERFORM SECURITY ACTIVITIES
PERFORM MOVEMENT SECURITY USING SMOKE Task 301
EXECUTE TRAVEL OVERWATCH Platoon Drill
MOVE USING WEDGE FORMATION Platoon Drill
MOVE USING LINE FORMATION Platoon Drill
DEFEND AGAINST AIR ATTACK Platoon Drill
EXECUTE ACTIONS ON CONTACT Task 147
EXECUTE ACTION LEFT (RIGHT) FRONT (REAR)
EXECUTE MOVE LEFT (RIGHT) REAR
PERFORM PLATOON FIRE DISTRIBUTION AND CONTROL Task 137
ESTIMATE OF THE SITUATION task 36
Individual tasks
MAINTAIN CONTACT WITH AN ENEMY Task 169
EXECUTE A PLATOON HASTY ATTACK Task 76
BY PASS AN ENEMY POSITION Task 172
communicate using visual sig
DRIVE TRACKED VEHICLES
REPORT
LOAD, ETC
ENGAGE TARGETS Task 107
ASSAULT AN ENEMY POSITION Task 173
MOVE ON SELECTED ROUTE USING OVERWATCH Platoon TECHNIQUES drill
TO INDIVIDUAL TASKS

FIGURE 25. Drill Diagram, with Drills Identified

a. Move
b. Attack
c. Defend

Each of these ARTEP Missions is comprised of other collective tasks besides "Move by Wedge Formation." Each of these other collective tasks are to be listed. Together, the ARTEP Mission Major Variants and Collective tasks provide a complete ARTEP-to-Drill task linkage and allow the author to work within the context of the entire MTP. This ensures that the trainer using the MTP is training within the MTP. If this is confusing, go back to Figure 2 and look once again at the cubes and data flow. What we are after here is accountability of all missions, variants, tasks, etc., and context (i.e., we train to move by wedge formations within the context of move, attack, defend ARTEP missions, variants, tasks, etc.. We also go from a wedge into an assault that requires a line formation, or we by-pass resistance which may require an echelon or a column). By always ensuring the integration of these tasks, the author can coordinate T&EOs, with STX, with drills, and illustrative Field Training Exercises (FTX), and have a coordinated Mission Training Plan that is ready for use.

Completion of the task integration phase prepares the author to write the drill. He does this by a combination of (1) writing the tasks, conditions, and standards right out of the CFEA data base, (2) illustrating the plan view of the location of each vehicle (or soldier for a dismounted "move in Wedge Formation" drill), (3) establishing the initiating cues, and (4) detailing the performance measures for each member of the platoon. These performance measures are derived from the individual tasks available from the CFEA. A wedge formation drill is shown in Table 5.

Developing Training Plans. Training plans are the plans that unit COMANDERS make to implement the MTPs in their units. These plans coordinate "How-to-Train" doctrine and "Train-to-Fight" programs by development of a concise training strategy that combines individual and leader training, drills and other collective tasks into STX. STX support unit conducted field Training Exercises (FTX), which, in turn, lead to unit mission (combat) proficiency. The training plan emphasizes (1) integration of tasks, (2) sustainment of proficiency, (3) decentralization of planning, and (4) multi-echeloned conduct of training. Decentralization means that the training plan provides flexibility for each level of training responsibility (e.g., squad, platoon, company, etc.) to develop plans unique to their requirements within the overall framework of the STX, drills, FTX, etc., which comprise the MTP. Multi-echelon refers to the preferred training method (i.e., within the context of normal roles and missions of each echelon). Here we expect the company command elements to train on command tasks while one or more platoons are conducting an STX/Drill. For example, one or more squads within a platoon may simultaneously conduct individual training, while other squads practice the collective tasks appropriate for the STX/Drill of the parent platoons.

Actual design of a training plan is a leader responsibility. The MTP provides the leader with sample plans illustrating design and training techniques, as well as samples of a training strategy. These sample plans provide the MTP user with two things:

TABLE 5

DRILL DESCRIPTION: WEDGE FORMATION

DESCRIPTION: The wedge formation is used for tactical cross-country movement. In this formation the tanks in the platoon move as shown in below and maintain their positions relative to each other using the wingman concept. The wedge formation permits excellent fire to the front and good fire to each flank.
REFERENCE: TT 71-1/2, Vol II.

DRILL TO ARTEP TASK LINKAGE: The wedge formation supports the following ARTEP 71-2 missions:

a. Move:
(1) Move in Bounding Overwatch (3-IV-1-9).
(2) Move in traveling (3-IV-1-7).
(3) Move in traveling overwatch (3-IV-1-8).
(4) Cross SP/LD (3-IV-1-3).
(5) Move (3-II-4-1).

b. Attack:
(1) Move (3-IV-2-1).
(2) Conduct a hasty attack (3-IV-2-8).
(3) Assault (3-IV-2-11).
(4) Conduct Passage of Lines Forward (3-IV-2-2).
(5) Bypass (3-IV-2-12).

c. Defend:
(1) Move (3-IV-2-1).
(2) Conduct Passage of Lines Rearward (3-IV-3-10).
(3) Conduct Fire and Movement (3-IV-3-8).

SPECIFIC PREREQUISITES: Knowledge of wingman concept.

HOW TO TRAIN DRILLS: See paragraph 3-7.

TABLE 5 (cont'd)

PERFORMANCE OBJECTIVES:

a. Task: Execute wedge formation.

b. Conditions:

(1) A tank platoon in any moving or stationary formation.
(2) Terrain suitable for movement into a wedge formation.
(3) An initial cue.

c. Standards:

(1) Platoon moves into a wedge formation without delay.
(2) Tanks are positioned generally as shown in Figure 3-7.
(3) Vehicle intervals are in accordance with platoon leader direction/unit SOP.
(4) Main guns are oriented in the same general direction as shown in below.
(5) Platoon maintains ground and air security.
(6) Platoon maintains radio-listening silence when in open hatch mode.

INITIATING CUES:

a. Hand and arm/flag signal (see Figure __).
b. Voice command (FM radio) - Platoon caTT sign:
"BRAVO 6 ROMEO - EXECUTE WEDGE."

TABLE 5 (cont'd)

PERFORMANCE STEPS:

(After PL gives initiating cue)

K 1	(PL	TANK 2	TANK 3 (PSG)	TANK 4
a.	Takes up the direction of travel.	Moves to the left rear of tank 1.	Moves to the right rear of tank 1.	Moves to the right rear of tank 3.
	*Orients the main gun toward the front.	*Orients the main gun toward the left front.	*Orients the main gun toward the front.	*Orients the main gun toward the right front.
b.	Maintains orientation on axis of advance. PL directs lateral dispersion.	Maintains lateral dispersion in accordance with PL direction/ unit SOP.	Maintains lateral dispersion in accordance with PL direction/ unit SOP.	Maintains lateral dispersion in accordance with PL direction/ unit SOP.
c.	*Maintains ground and air security.	*Maintains ground and air security.	*Maintains ground and air security.	*Maintains ground and air security.
	*Maintains radio-listening silence.	*Maintains radio-listening silence.	*Maintains radio-listening silence.	*Maintains radio-listening silence.

*This action must be performed throughout the drill.

1. A set of plans that a typical unit can follow to reach mission proficiency.

2. A departure point for unit development of training plans based on unique unit missions, personnel conditions, terrain and facilities.

A training should be designed to cover a period of from one to six months. That is, it should represent the amount of collective training time available to a unit within one to six months. Sample Training plans are shown at Appedix H. These are drawn from TRADOC Circular 310-2 (TEST) and are provided here for ease of reference. The MTP author develops these samples by:

1. Reviewing the CFEA and STX/ Drills to isolate groups of tasks or missions that, when trained to proficiency, are most likely to result in the most combat capability. It is this criterion that will most likely cause a unit commander to assign top training priority. In reality, this is primarily a function of how often a collective task appears across the range of missions. "Movement" is a good example. "Shoot" is another. Together, they constitute the core group of collective tasks essential to "Fire and Movement."

2. Developing a FTX in which the unit expected to actually execute and evaluate its proficiency in the selected groups of tasks. (The characteristics and developing steps of an FTX are discussed in the next paragraph.)

In developing the sample Training Plan, the MTP author should ask and answer the following questions:

1. Which tasks require collective practice on the part of troops?

2. What are the standards and performance measures that apply to unit leaders in the excution of these collective tasks?

3. Which unit functions (movement, fire and manuver, sustaining operations, defend a battle positions, defend urban terrain, etc.) are supported by the selected collective task? The plan should contain elements of each unit function.

4. Which of the ARTEP Missions and Variants do each of the ARTEP tasks and collective tasks selected for inclusion in the training plan support?

5. Which actions/standards can be meaningfully trained only in the context of a collective exercise?

6. Which tasks must be mastered, in whole or in part, before a unit can train to the next higher or adjacent (vertical/horizontal) task?

7. What training resources are required to execute the training plan? (Resources are developed for each exercise in the plan.)

A mission training plan diagram for the Mission of "Attack" is shown in Figure 26.

Developing the FTX. Field Training Exercises (FTX) are groups of STX, which contain the drills, placed into a tactical scenario by the addition of an OPORD/FRAGO, and tied to specific training objectives and resources. FTX are not prescriptive, as are drills, but are descriptive in that the unit is free to execute them as designed or to use them as a "template" to design FTX for unit unique requirements STX and drills and individual tasks are mandatory.

The FTX is part of the Mission Training Plan. Illustrative examples of the FTX are prepared by the MTP author, as discussed previously in the section on development of training plans. In fact, Figure 10 diagrams the essential element of an FTX for the Hasty and Deliberate Attack. Although the Night Attack is included in the diagram for completeness, the major variants that make the Night Attack distinct from the deliberate/hasty attack were omitted from the example. Notice that the major variants that comprise the Hasty Attack (i.e., movement to contact, breach obstacles, and tactical road march) have already been developed as STX. The same is true for the major variants of the Deliberate Attack (i.e., the passage of lines, and establishment of near and far bases of fire). (STX will also have been developed for the Night Attack by the time the MTP author reaches development of FTXs).

Each FTX consists of:

1. An FTX diagram similar to the one in Figure 10.

2. A list of STX that make up the FTX.

3. A list of drills supporting each STX (this list is already a part of the STX).

4. Resource requirements:

   a. Maneuver area recommended as required to perform the STX and support additional actions the MTP author seems necessary. An example of an additional FTX action would be the training of the battalion tactical operations center and the battalion support platoon within the conduct of a company/team FTX. In this case, the example FTX would display additional resource requirements.

   b. OPFOR recommended as required to provide the needed enemy for each STX.

   c. Communication, command and control (C3) as required.

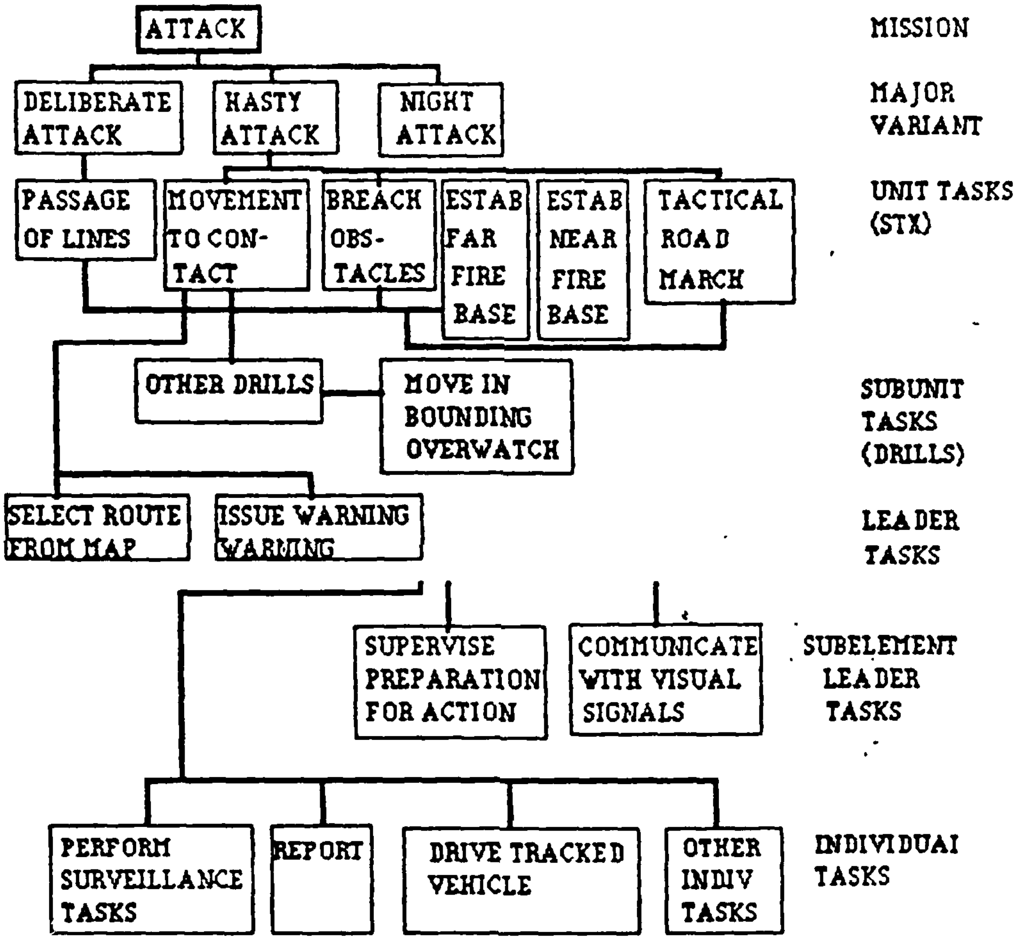


FIGURE 26. Training Plan Mission-Task Diagram

d. Training devices such as the laser engagement systems, training mines, radio jamming devices, etc.

e. Ammunition, to include blank rounds, pyrotechnics, and live ammunition if a live fire exercise is included.

5. Unit requirements expressed as task organization (i.e., two tank platoons, one mechanized infantry platoon, one combat engineer squad, a fire support team, mess team, maintenance team, etc. for a company/team FTX).

6. Conditions. The general tactical conditions that required the company (Platoon?) to execute the mission, in this case the Hasty Attack.

7. The Operations Order (OPORD). An OPORD or Fragmentary Order (FRAGO) contains additional conditions. The OPORD/FRAGO are based on Army doctrine and are prepared in a specific format with detailed information needed for the conduct of the exercise. The standard OPORD contains five paragraphs:

   1) Situation
      a. Enemy Forces
      b. Friendly Forces

   2) Mission Statement

   3) Execution of the operation that details a specific tactical mission for each participating subordinated unit.

   4) Service Support

   5) Command and Signal

To sum up, the MTP author depends mainly on previously developed materials (STX, Drills, CFEA) and analysis to develop FTX. Chapter 5 of the Armor School's TC 17-15-1 contains detailed FTX examples for each major tank platoon mission.

Developing MTP Common Modules Common missions and tasks are those that apply to Army units in general or are required by other TRADOC schools. As a result of incorporating these common missions and tasks into the MTP the author must have a listing of modules he receives and of those he is responsible for preparing for other schools. When preparing common modules the MTP author must ensure application of procedures specified in TRADOC Regulation 310-2, even though his own school (in the case of CAPS that means Infantry and Armor Schools), may specify unique procedures or formats.

TRADOC Regulation 310-2 requires the preparing school to submit the common module to the Combined Arms Center at Fort Leavenworth for approval. Once approved, the Infantry School (and the Armor School as appropriate) will publish the module and provide it to all other MTP preparing agencies.

The common modules are prepared in the same manner as other T&EOs. Common modules received from other schools may be modified to suit the needs of the recipient school.

Developing the Training Matrix. The Training Matrix relates the training products (STX and Drills) to the Mission Training Plan (MTP) and associated FTX, as well as to the individual soldier's and leader's tasks. For this reason, it is possible to view the training matrices as "roadmaps" to MTP development. Properly developed, these matrices chart the STX that must be mastered prior to conduct of a given FTX and the drills, individual soldier tasks and leader tasks which must be trained prior to the STX.

TRADOC Regulation 310-2 provides the approved format for the training matrix. However, both the Armor and Infantry Schools have developed slightly different formats. All formats contain the same data, which are enumerated below:

a. The FTX-to-STX matrix relates the STX to the FTX, showing vertically the STX that support a given FTX, and showing horizontally the commonality of a specific STX in the various FTX.

b. The Drill-to-STX matrix depicts drills that support a specific STX. Drills are listed in the vertical column and the commonality of a given drill to the STX is shown in the horizontal column.

c. The Leader Task-to-STX matrix shows the relationship between various STX and supporting leader tasks. As in the other matrices, supporting relationships and commonality are shown by the column and row arrangement.

d. The Individual Task-to-STX matrix displays those tasks not trained in specific drills but essential to the STX, such as "posting maps" and "maintaining journals." Format is the same as other matrices.

e. The Indivdual Task-to-Drill matrix shows the ultimate Soldier's Manual task relationship to the drills and STX.

Because each Mission, collective task, individual leader task, and soldier task has a task statement assigned that contains common descriptives (e.g., attack, move, defend, surveillance, reconnoiter, etc.), and because each STX is given a letter designator, it should be possible to formulate the CAPS database management system around these matrices. In fact, the matrices have been developed to allow both the MTP author and the unit trainer to manage the paper-based manual data system. Remember, the FTX-to-STX matrix is drawn after the training plans are produced. Other matrices are drawn directly from the CFEA and MTP development process. TRADOC Regulation 310-2 (TEST) and the Armor School Mission Training Plans contain examples of the matrices discussed above.

The Infantry School's Individual Task-to-Drill matrix and Platoon Mission-to-Drill matrix (not required by TRADOC Regulation) are attached as Appendix F.

Developing the Unit Test. The Unit Test is a performance oriented test to be administered to squads, sections, platoons, and company/teams by the parent headquarters two echelons above the tested unit.
(A section is two or more squads and is unique to scout platoons, mortar platoons and support elements of the Mechanized Infantry and tank battalions. The infantry and tank platoon themselves do not organize in sections. Tank platoons are organized by crew and infantry platoon are organized by squads) The unit test is designed to assess the tactical proficiency of the unit, using a combination of T&EO selected from the MTP against a realistic OPFOR. All testing is done with the use of the Multiple Intergrated Laser Engagement Systems (MILES).

The Unit Test is organized into six sections, as follows: 1) Introduction, 2) How to Manage the Test, 3) Instruction as to test preparation, 4) Conduct of the Test instructions, 5) the After Action Review and 6) the Test Scoring System.

TRADOC Regulation 310-2 (Test), (Appendices A thru G), presents a detailed discussion of the development of the unit test. The reader is advised to use this source reference as the primary developmental tool for automating the Unit Test module of the CAPS database. The sample unit test presented in TRADOC Regulation 310-2, beginning on page 86 and ending on page 123, contains the data/information required for the MTP author at the TRADOC School to develop a publishable unit test with minor revisions to reflect command guidance. For this reason no further discussion of unit test development will be presented here.

Preparing the Mission Training Plan. Design and preparation of the Mission Training Plan (MTP), including format, organization, and content, is governed by Chapter 9 of TRADOC Regulation 310-2, Development, Preparation, and Management of Army Training and Evaluation Program. This Chapter is really a presentation of the adminstrative organization, format, and content of the MTP.

Comparison of Infantry and Armor School Formats with TRADOC Regulation 310-2

USAIS and TRADOC Regulation 310-2 Squad and Platoon MTP Contents. In as much as the Infantry School MTP organization, format, and content varies from the regulatory example, the reader is advised to read the regulation from the standpoint of the CAPS Study, to wit: The CAPS design will provide the capability to implement the requirements inherent in TRADOC Regulation 310-2, while retaining the flexibility to implement variations required to allow the MTP author to comply with changes dictated by local (Infantry School) requirements and command guidance. At this stage of analysis, this is taken to mean that the database organization (i.e., the full range of training matrices) will be embedded in the database management system, but that the author will be provided with a range of implementation options as to formats and data sorts, without the need to re-program CAPS software. TRADOC Regulation 310-2 MTP organization is shown in Figure 27.

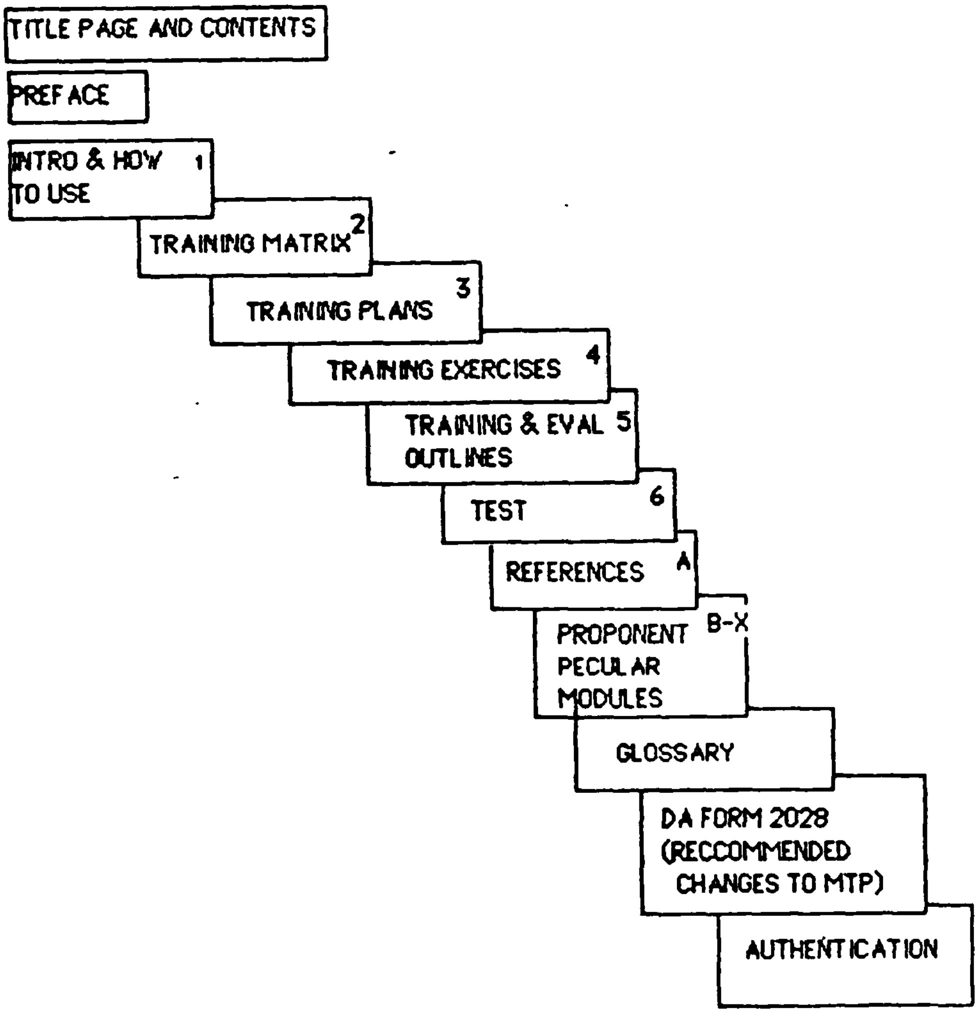


FIGURE 27. Mission Training Plan Organization

The comparison shown in Table 6 illustrates the differences between Infantry School (USAIS) Squad and Platoon MTP formats and the requirements of TRADOC Regulation 310-2. Infantry School company and battalion MTP content and formats are presented in following paragraphs. These format and content difference are the result of command perspective difference and are not amenable to solution by the CAPS automation designers. For this reason, all parties have agreed to design CAPS to the 310-2 universe with sufficient data sort and format flexibility to allow the local MTP author to respond to local requirements.

US Army Armor School Platoon MTP Form/Contents. The Armor School's Platoon MTP more closely follows the TRADOC Regulation. The Platoon MTP format is similar to the company and battalion formats. However, the format shown below, although current, is not the same as the one shown previously in Figure 8. Information available indicates that the Armor School plans to bring their MTP products in line with 310-2 during the next revision cycle.

ARTEP 71-2 Company Level Format Outline. The Infantry School has not finalized the format and content of the MTP because authorities are still evaluating the suitability of TRADOC Regulation 310-2 to meet the needs of the Infantry community. The format outline shown in Table 7 represents thinking at the school at the time this report was researched (November 1984).

Mission Training Plans/FTX and Unit Tests as required by TRADOC Regulations 310-2 can be developed from the T&EO and Training Matrices listed above if the underlying CAPS data base is organized to assist the MTP authors at the Infantry School. It is understood that the reference, glossary common modules, and resources required to execute the STX/FTX and drills will be included in the Infantry School MTP. Coordination with the Armor School reveals that their plans for MTP development conform closely to the standards promulgated in TRADOC Reg. 310-2.

ARTEP 71-2 Battalion Level Format Outline. The Infantry School's planned Battalion/Task Force Mission Training Plan format and content is shown in Table 8. Comments relating to the comparison of TRADOC Reg 310-2 and Infantry School plans set forth in connection with the company level Mission Training Plan apply to this document also.

US Army Infantry & Armor School MTP Development Organizations

Overview. The Infantry (USAIS) and the Armor Schools (USAARMS) are organized differently with regard to the production of doctrine, tactics, and Mission Training Plans (MTP). The purpose of this paragraph is to establish the MTP production processes as they now exist at each school.

The two schools differ their organization for MTP development primarily because of the units that must be supported by the respective schools. The Infantry School must develop doctrine, tactics, and MTP for airborne, airmobile, light mechanized infantry, and Rangers, while the Armor School is charged with developing these products for armor and cavalry only. Personnel situations and these responsibility differences have resulted in the different organizations.

TABLE 6

MTP FORMAT AND ORGANIZATION
COMPARISON BETWEEN USAIS AND TRADOC REG 310-2

USAIS	TRADOC REG 310-2
Chapter 1 INTRODUCTION	
Basically the same w/ some minor exceptions .Added discussion of triangle .Diffs in How to Use MTP in Chap 1-6 due to diff in Chap.	1-1 General 1-2 Contents of the Program 1-3 Supporting Materials 1-4 Basic Idea/Missions 1-4 Training Principles 1-6 How to Use MTP 1-7 Training Evalutions
Chapter 2 TRAINING MATRIX	
.Mission-to-Drill Matrix .Mission-to-Supp Task Matrix .Drill-to-Individual Task Matrix	.FTX-to-STX Matrix .STX-to-Drill Matrix .STX-to-Leader Task Matrix .STX-to-Separate Task Matrix .Drill-to-Indiv Task Matrix
Chapter 3 HOW TO TRAIN	TRAINING PLANS (HANDOUT)
3-1 Introduction 3-2 Planning Prep Tips 3-3 Execution Tips (Illus using MSN-Move to Contact-Hasty Atk)	Diagrams Relating STX Required to Support an FTX
Chapter 4 (Non Comparable Para)	TRAINING EXERCISE
	4-1 Details CO FTX Supported by Training Plans 4-2 Details PLT STX Supported by Training Plans

TABLE 6 (cont'd)

Chapter 4 (USAIS only) T&EO (Handouts)	Chapter 5 T&EO (Handouts)
Element	Element
Task (Drills, C3 Function or Other) Task (More General Actions/Standards	Mission Conditions Subtask Standards Standard Number
3 Column/Iterations STD Rating	5 Column/Iteration STD Rating
Evaluate W/Check Mark	Evaluate W/ GO or NO GO
Task Performance Summary (No Percentages)	Task Performance Summary (W/Percentages)
Chapter 5 EVAL/ARTEP UNIT TEST	Chapter 6 TEST (ARUT)
Appendices	
A. Sample Test Plan B. Sample STX C. Sample OPORD D. Mission Planning Considerations	A. References
Glossary	Glossary

TABLE 7
COMPANY LEVEL MTP FORMAT

Chapter 1	Introduction
Chapter 2	How to Train - Individual Tasks - Collective Tasks - Drills - Drills-to-STX Matrix - STX-to-Mission Matrix
Chapter 3 (T&EO)	Company/Team Missions - Operate Command Post - Move - Attack - Defend
Chapter 4 (T&EO)	Company/Team Supplemental Missions - Occupy Assembly Area - Conduct Passage of Lines (FWD) - Conduct Passage of Lines (BACKWARD) - Conduct Breakout from Encirclement - Conduct relief in Place
Chapter 5 (T&EO)	Specialized Company/Team and PLT Missions - Conduct a Raid (Mounted) - Conduct a Raid (Air Assault or Dismounted) - Defend an Urban Area - Conduct Deliberate Attack of Urban Area - Infiltration - Hasty River Crossing

TABLE 8

BATTALION LEVEL MTP FORMAT

Chapter 1	Introduction
Chapter 2	Training the Battalion - Individual Tasks (Staff) (Skill Level 1-4 and MQS III Functions) - Collective Tasks (by Staff Section) - Drills (TOC, S2-3, S1-4) - Map Exercise (MAPEX) - Tactical Exercise w/o Troops (TEWT) - Battalion Collective Training - FTX
Chapter 3 (T&EO)	Battalion Missions - Operate the TOC, MOVE, ATTACK, ETC,.
Chapter 4 (T&EO)	Battalion Supplemental Missions - Conduct a Strategic Deployment - Defend Urban Area - Conduct Deliberate Attack in Urban Area - Conduct Air Assault - Conduct Hasty River Crossing - Conduct Deliberate River Crossing - Follow and Support
Chapter 5 (T&EO)	BN/TF and CO/TM Supplemental Missions - Occupy Assembly Area - Conduct Passage of Lines (FWD) - Conduct Passage of Lines (BKWD) - Conduct Breakout of Encirclement - Conduct Relief in Place
Chapter 6 (T&EO)	BN TF and CO/TM/PLT Supplemental Missions - Conduct a Raid (Mounted) - Conduct a Raid (Air Assault or Dismounted) - Conduct a Demolition Guard Force Mission - Secure Atomic Demoliton Munitons - Conduct Area Protection/Area Damage Control Operations
Glossary/References/Resources,etc,.	

The Infantry School Organization. The Infantry School organization for MTP development is shown at Figure 28. As shown, the Directorate of Training and Doctrine is charged with developing the CFEA, individual Soldier's Manuals, Military Qualification Skills (MQS) manuals (the MQS is to the officer what the Soldier's Manual is the the enlisted man). Additionally, the DOTD develops the squad and platoon MTP. The Combined Arms Tactics Directorate (CATD) develops the infantry doctrine for squad thru brigades and authors the company and battalion MTP. The Ranger Department develops all Ranger MTP, from squad to battalion. The 29th Infantry Regiment is an Infantry School Training Regiment that develops and teaches all infantry-unique maintenance tasks.

The interaction to develop CFEA products at the Infantry School is between the Tactics Divisions of CATD and the Analysis and Studies office of DOTD. Unit missions and tasks (from highest to lowest echelon), and the associated collective task analytic information are developed by the CFEA analysis within the Analysis and Studies Office of DOTD, in consultation with CATD, and are entered in the CFEA database. This database is then used by writers in both DOTD (squad/platoon) and CATD (company/battalion) to develop the T & EO, Drills, STX, and FTX of the MTP.

Appendix D, Work Station Data Flow Matrices, details how the two organizations at the Infantry School exchange data in the process of developing the MTP. Essentially, the DOTO Collective Training Branch develops the products shown previously in Table 6. The A & B Branches of Doctrine Division of CATD develop the company and battalion MTP, outlined previously in Tables 7 and 8.

The Armor School Organization. USAARMS organization for MTP-production is as shown in Figure 29. The DOTD is charged with conduct of the CFEA, and development of Soldier's Manuals, MQS, and Armor doctrine. Development of tactics, doctrinal, and tactical instruction, and all MTP development is a responsibility of the Command and Staff Department (C&SD).

SUMMARY AND CONCLUSIONS

Impacts of differences in ARTEP Production Organizations. Our analysis of the differences between the ARTEP-production organizations at the Armor and Infantry Schools, and between these organizations and the requirements of TRADOC regulations, indicate that these differences are minor with respect to the impact on any automated ARTEP-production system that might be designed. The database can, and should in any case, contain all matrices, missions, subtasks, and relational data to develop drills, STX, FTX, training plans, etc., as desired.

Conclusion: Automation of the MTP production process will provide flexibility to allow many variations in format and content that may need to be produced, with minimum effort. This will allow testing validation in field of various MTP formats, as an aid in determining the most effective developmental strategy.

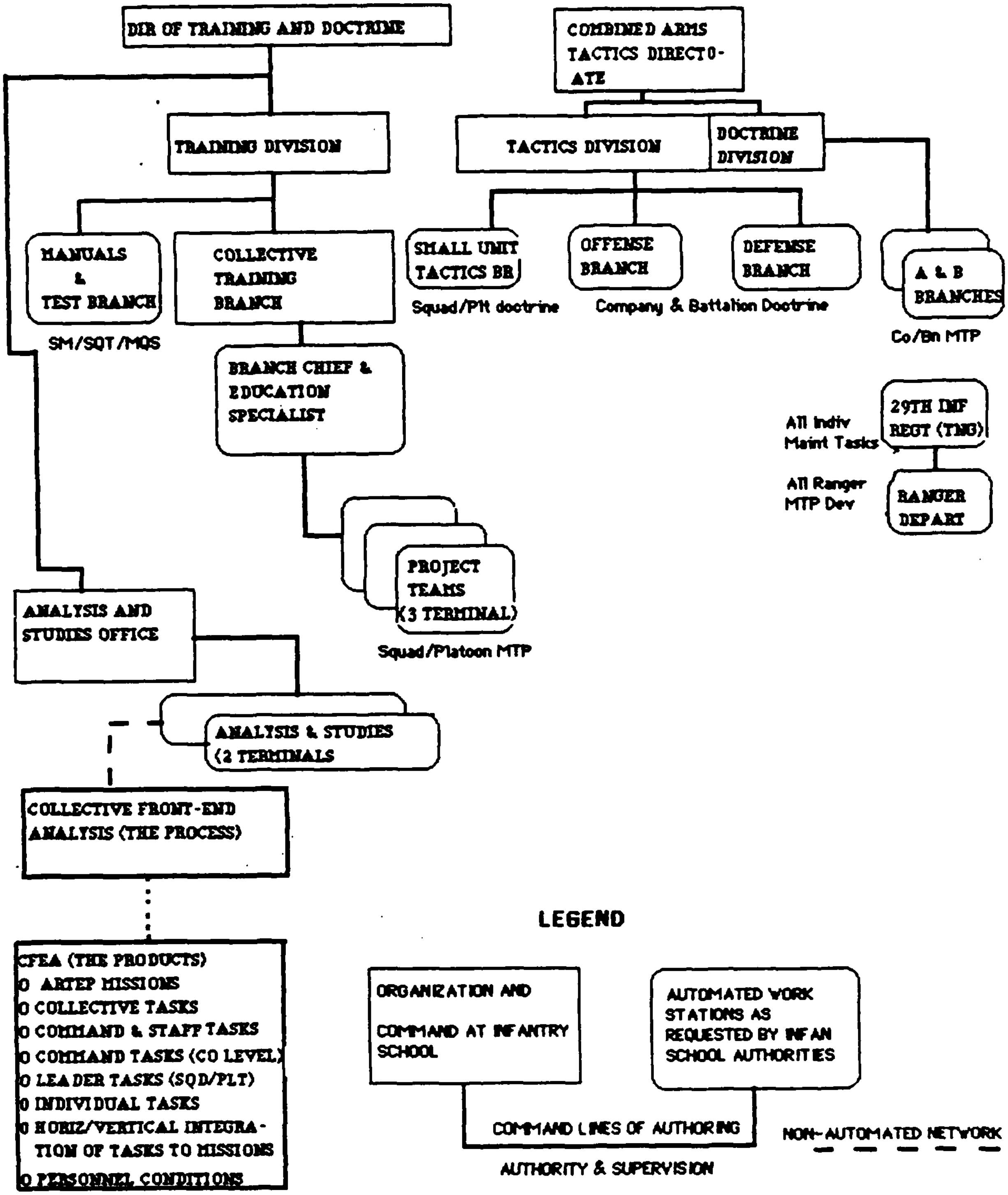
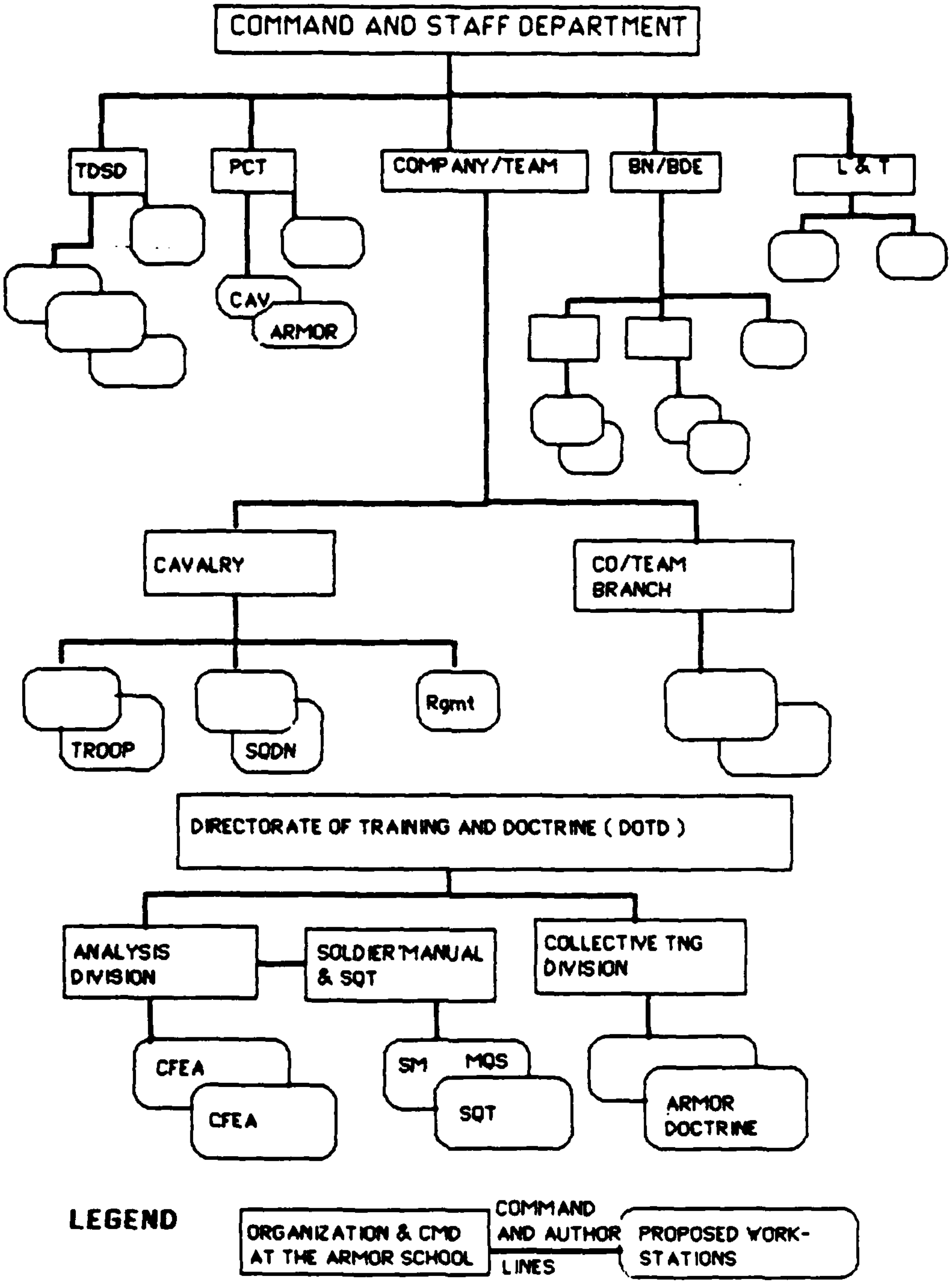

DIR OF TRAINING AND DOCTRINE
COMBINED ARMS TACTICS DIRECTO-ATE
TRAINING DIVISION
TACTICS DIVISION
DOCTRINE DIVISION
MANUALS & TEST BRANCH
SM/SQT/MQS
COLLECTIVE TRAINING BRANCH
SMALL UNIT TACTICS BR
Squad/Plt doctrine
OFFENSE BRANCH
DEFENSE BRANCH
Company & Battalion Doctrine
A & B BRANCHES
Co/Bn MTP
BRANCH CHIEF & EDUCATION SPECIALIST
All Indiv Maint Tasks
29TH INF REGT (TNG)
All Ranger MTP Dev
RANGER DEPART
PROJECT TEAMS (3 TERMINAL)
Squad/Platoon MTP
ANALYSIS AND STUDIES OFFICE
ANALYSIS & STUDIES (2 TERMINALS
COLLECTIVE FRONT-END ANALYSIS (THE PROCESS)
CFEA (THE PRODUCTS)
0 ARTEP MISSIONS
0 COLLECTIVE TASKS
0 COMMAND & STAFF TASKS
0 COMMAND TASKS (CO LEVEL)
0 LEADER TASKS (SQD/PLT)
0 INDIVIDUAL TASKS
0 HORIZ/VERTICAL INTEGRATION OF TASKS TO MISSIONS
0 PERSONNEL CONDITIONS
LEGEND
ORGANIZATION AND COMMAND AT INFANTRY SCHOOL
AUTOMATED WORK STATIONS AS REQUESTED BY INFAN SCHOOL AUTHORITIES
COMMAND LINES OF AUTHORING
AUTHORITY & SUPERVISION
NON-AUTOMATED NETWORK

FIGURE 28. Infantry School MTP-Development Organization



TDSD=Training Development Support Division
PCT= Platoon & Compnay/Team
L & T= Literature & Test
Rgmt= Regiment
BDE=Brigade

FIGURE 29. Armor School MTP-Development Organization

Desirable Characteristics of ARTEP-Production Automation. It is useful to discuss how CAPS might help authors in terms of the general decision tasks discussed above. The following capabilities for an automated ARTEP-production system seem to offer the best return on the automation investment:

1. Automation should guide authors in analyzing training-relevant features of collective tasks such as gunnery, map reading, movement techniques, sustaining tasks, and supervision tasks. Analysis in terms of individual soldier's leader and command tasks may be very fruitful in this area.

2. Automation should record results of the above analysis in the database in a manner that is transparent to the author.

3. Automation should support the application of any improved ARTEP concept. In other words, the system should provide a complete set of training-relevant information about collective tasks and ARTEP missions, independent of the ARTEP format desired. Furthermore, it should allow the creation and modification of formats by authors.

4. The relational nature of the database, database management system, and the user interface(s) should be such that the author can (1) selectively draw upon types of information that apply to a particular MTP design concept, (2) experiment with new designs, or (3) change designs or formats to enhance acceptability or usability in their field.

5. Automation should include rules for developing MTP components, such as drills/STX, which can be specified in terms of training-relevant task features and relationships. For example, the identification of sample STX/FTX and their incorporation within a descriptive unit training plan, will require consideration of task features, such as echelon, type of individual and collective tasks, etc.

6. CAPS should be able to identify, sort, and manipulate tasks in its database, based on the following attributes:

   a. Task does/does not require collective practice on the part of troops.

   b. Identification of actions/standards which apply to unit leaders only.

   c. Identification of standards/actions that are better trained outside the context of collective training (e.g., individual skills which, if trained in a collective exercise, would distract leaders attention from the remainder of the unit for too long a period).

d. Identification of a "unit function" that is served by each task, such as movement, sustaining operations, conduct of fire, and occupation of battle positions.

e. Identification of training resource requirements and estimates of the amount of each resource such as ammunition, fuel, evaluators, time, ranges, devices.

f. Identification of Soldier's Manual tasks that are performed, in whole or in part, during the conduct of the task.

g. Identification of Soldier's Manual tasks that are prerequisites to training (i.e., unless tasks have been mastered, the unit is not ready to train the collective task).

h. Identification of actions/standards that can be meaningfully trained only in the context of a collective exercise.

i. Identification of the ARTEP mission to which each task (collective and individual) applies.

j. Identification of tasks that must be performed, in whole or in part, to inform, organize, equip, and/or position a unit to train to the next higher or successive task.

In short CAPS should go beyond automating the manual system; it should instead take advantage of automation's power to make a better product with less effort.

FUNCTIONAL REQUIREMENTS OF CAPS

Based on the analysis of the exisiting ARTEP production process, we have identified a number of functional capabilities that must be included in a system that would provide computer aiding for the ARTEP production process. Such capabilities include automated database management, electronic information distribution, and integrated word and graphics processing, among others.

Idea Processing Requirements

Before describing the requirements for specific functional capabilities in CAPS, it is important to highlight the central requirement for any computer aids for the ARTEP production process, namely the requirement for idea processing. Although it seems obvious to say it, the primary focus of any process like the production of ARTEP materials is the creation, communication, and coordination of ideas. Thus, any sysem that aids the ARTEP process must assist the users in dealing with concepts, as well as words and graphics. Whereas most computer-based systems will include some form of word processing capability, this does not necessarily provide the tools that are needed for creating and manipulating ideas, which may be ill-formed and only partially expressed; well-stated but loosely organized; or highly complex; well organized; and multi-dimensional. In all of these cases, ideas or concepts are more than a collection of words. Thus, an idea processor must include aids allos the user to manipulate ideas, and not be constrained to attend to the particular requirments of the compuer aid itself.

Several attributes that a computer-aided ARTEP production system must possess include the following:

- The system controls the flow of work;
- The system aids the users by providing proper templates for performance of tasks;
- The system provides database search utilities and design aids to assist the users to complete task templates;
- User entries consist of concept descriptions (text and graphics) and structural linkages;
- The system automatically extracts any required process implications;
- The system automatically files the results of the user's work locally, and distributes the resuls globally as appropriate;
- The system automatically generates and propagates authoring/revision assignments through the product authoring hierarchy.

Author- and Decision-Aiding Utilities

The objective of this section is to address new concepts for the CAPS system design that have come out of recent efforts to provide intelligent aids in instructional authoring and other complex problem-solving tasks.

Intelligent Authoring Systems. The availability of inexpensive computing power has increased the interest in tools for creating computer-based training materials. Most of the authoring systems created to date have concentrated on making the programming task easier for the user by means of simplified authoring languages. Computer programming is only one component of the authoring task and important only when the delivery of the instructional materials is to be via computer based instruction. Perhaps the more difficult part of the authoring task is instructional design decisions required at each stage of the instructional design process. Recent efforts have begun to concentrate on developing intelligent authoring systems that assist in the instructional process per se and not just on the programming part of the task. Such systems are identified as Computer-Aided Instructional Design Systems (CAIDS).

A computer aided instructional design system is a series of intelligent instructional design tutor/editors which provides two kinds of assistance: First, a CAID system helps authors with minimum instructional design experience to prepare effective instructional materials. Second, a CAID system helps authors with considerable instructional design experience to create large amounts of effective instructional materials for less cost than is possible without such computer based assistance.

CAIDS provide assistance in a number of ways, including:

Information Processing. The word processing and database capabilities of the computer facilitate the processing of large volumes of information eliminating the necessity to reenter redundant information. Second, the database management system can direct the information in a complex system to the appropriate authors at the appropriate time thus eliminating the need to "reinvent the wheel" at various stages in the instructional design process.

For large projects, these clerical functions alone would justify the use of a computer-based instructional design system. However, CAIDS editors can also provide a certain amount of intelligence to further enhance the instructional design and development process. Several types of intelligence could be incorporated.

Guided Input. Input templates could be enhanced via guided input of information. Guided input consists of a series of structured decision trees where each node is presented via a question to the user. For the naive system user these questions guide the decision process leading to the input that is necessary for a given instructional design product. Such guidances makes it unlikely that the user will fail to consider important data necessary to the decision or skip over information that may seem obvious but which is necessary to later stages in the development or training process. Guided input is especially valuable for the inexperienced user by providing necessary guidance in making instructional decisions. For the experienced user, such guidance gets in the way of efficient system use and slows the input of relevant information. The advantage of guided input as an option is that is is available for the inexperienced user but can be ignored by the experienced user.

Hypertext Tutorials. Input templates could be enhanced via hypertext tutorials. Hypertext is information that can be accessed by selecting any technical term in the input template and having the computer screen provide a tutorial overlay concerning this text. Tutorial overlays would supply information such as the definition of the term, examples of the term, relationship of the term to other terms, syntax information regarding the input, and other information which expands the information requested. The user merely selects the term or phrase for which more information is requested and selects tutorial from the menu to expand the term or phrase into the level of tutorial information necessary to understand the input template. Hypertext is especially useful for the inexperienced user but also provides a valuable reference for the experienced user especially for seldom used templates or infrequently required information.

Consistency Checker. User input can be subjected to a set of consistency checkers that ensure that the input in one phase of the instructional design process is consistent with previous instructional decisions. Such checkers can also ensure that the input is consistent with the expected information for a given component of the input template. Instructional design documents must be subject to certain regulations concerning the terminology to be used to describe specific events, conditions, standards, criteria, etc. Frequently designers change the conventions as they proceed through the instructional development process. Such changes in convention cause confusion for others who must use the interm design products in later stages of the design process. A consistency checker would flag such inconsistency of convention and ask the user to use accepted conventions for inputing the necessary data.

Input Translators. If design documents are to communicate to the wide range of users in a complex instructional development system, it is important that they have a consistency of form and output to facilitate this communication. Whenever consistent forms are used, a certain amount of this is "boiler plate" that is necessary for communication, but which requires considerable excess effort on the part of the designer to prepare. The necessary input often consists of a few terms for phrases which must then be embedded in the standard form for presentation. A CAID system would contain a number of translators that could take given input information and convert it to a number of different output formats as may be required by the given design task under consideration. In this way, the authoring is made considerably more efficient since the user must only input the necessary data the system would supply the "wrap around" necessary to convert this data into an output format appropriate for a given document.

System Issues Addressed by Decision-Aiding Concepts. In addition to the assistance that can be provided with the various editors within a computer-aided instructional design system, there are a number of decision aiding concepts that have been developed within the context of artificial intelligence. These various decision aids have direct applicability to the Computer-Aided ARTEP Production System.

Knowledge Representation. Knowledge representation is of interest in CAPS because of the need to capture the expert behavior and knowledge embodied in the ARTEP authoring process. Should CAPS be able to capture and represent this knowledge, it could be used to provide a basis for embedded training, automated procedures and aids to procedures, self-diagnosis of authoring and content errors, and more effective structuring of databases.

Adaptivity. One of the primary reasons for the ineffectiveness of the current ARTEP production process is its inability to adapt dynamically to the changing equipment, threat, doctrine, and operating environment. The long production cycle for the ARTEP creates a lag between training needs and ARTEP content. This time delay and lack of provision for feedback into the ARTEP production process promotes the carryover of inaccuracies and creates built-in obsolescence in the training product.

An automated system that was not readily adaptable to these same changes would be no better than the current manual system. Many computer-based systems have been even more rigid than their manual counterparts because of the inertia created by fixed-format data structures, complex and difficult to learn procedures, and inflexible and difficult to modify software. Such inflexible architectures have tyrannized many an attempt to modernize an existing data processing system.

CAPS can help to avoid the pitfalls of rigid system designs by building the flexibility and procedures for change into the initial system architecture. The precursors of this type of flexible architecture have evolved from attempts to model the flexibility of human problem solving. The human problem solving process is quite flexible under conditions in which there is available information and the human is not over-stressed by time pressure or uncertainty.

The CAPS design must provide mechanisms for using feedback from the field. A means must be provided for predicting the effectiveness of the mission training plans and for measuring that effectiveness. These effectiveness measures must be then related back to the authoring process and to the steps that must be taken to correct the training plans. The CAPS products should converge rapidly on workable and validated training approaches and should have a minimal and predictable lag with current technology and doctrine.

Guidelines for how to approach the adaptivity problem can be derived from new approaches that have been taken to the development of aids to intelligence analysis (Thompson, et al, 1983; Preprint, 1984). These new approaches recognize the need to consider individual skill differences in users along lines of ADP skill as well as domain knowledge and specific experience in performing given tasks.

As with many other military systems, ARTEP production is faced with a constant turnover of personnel, many times with insufficient overlap for effective transfer of institutional knowledge to incoming personnel. An automated system can aggrevate this problem if it requires extensive ADP skills on top of other job requirements.

Embedded Training. A capability with potentially high payoff for CAPS is to provide some degree of embedded training in the interactive user dialog. The projected payoff of this capability would appear in terms of:

- Elimination of pre-training for ARTEP authors;
- Fewer user errors;
- Compensation for variation in user skill levels in ADP, task, and domain knowledge;
- Increased versatility of users in authoring and document production tasks.

Ease of Use. Embedded training by itself would not necessarily make CAPS easy to use. Other factors can have a great impact on whether the user interface is a "friendly" interface or will have features that make it difficult to learn and use, less productive in accomplishing authoring tasks, and more prone to user errors. New hardware technology in intelligent workstations and new concepts for interaction using multiple window displays and pointing mechanisms for user inputs have promoted a new style of interactive dialog that is much more versatile and powerful for complex computer-aided tasks. This new style of interactive dialog has solved some major problems in interaction with general purpose computers (as typified by current Apple Macintosh software). However, these new capabilities have not been carried into complex knowledge-based tasks that characterizes the requirements of CAPS.

Related Decision-Aiding System Developments. One of the problems of the ARTEP production process is that many small decisions go into the authoring process that are transparent to the novice and often-times to the system designer. These decision processes are transparent when the expert performs them because they have become automatic and integrated into the job. The novice has difficulty in performing these decision processes because they are not articulated in standard operating procedures nor easily taught by the expert to the novice. The novice is often faced with the prospect of making bad decisions or with having his work redone by someone more senior. Such types of decisions would occur in situations where the author must decide whether to investigate the need for changing particular training plan sections on the basis of new requirements, feedback from field units, or changes in source documents. The decision process of the expert will be based on in-depth background knowledge and experience with analogous situations. Because the novice has no such experience, he cannot be expected to make the proper inferences.

The potential use of decision-aiding mechanisms to support the authoring process is based on having performed prerequisite analysis and representation of the knowledge components that go into specific authoring processes and being able to distinguish the characteristics of expert performances. In certain cases, in which there is no actual expertise, such as when a totally new mission or weapon system is introduced, there is the potential for an authoring aid to prescribe a structured decision process to support the authoring task.

Knowledge Engineering. Knowledge engineering is an emerging expertise that has developed from the need to understand and encode the knowledge components of computer-aided expert systems. Knowledge engineering has derived knowledge representation schemes from research work in artificial intelligence and current applications of these theories in building expert systems for specific domains. Perceptronics has been involved in the applications of AI to building aids for tactical planning at the Corps level, intelligent aids for Tank Platoon route planning and target intercept, battlefield threat modeling, tactical operation generation, and battlefield aids. Knowledge engineering in these instances has been required to bridge the gap between computer science representations of problem solving and military science representations of tactical problem solving.

The knowledge engineering for CAPS has been performed to a great extent by this program and the results of that work are contained in this document. This work has not, however, carried the representation of authoring knowledge all the way to software specification form.

Selection of AI Paradigm for CAPS. The paradigm that determines the general design philosophy for the development of authoring aids in CAPS must be carefully selected on the basis of matching available technology, user skills, performance goals, and development resources to the authoring problem.

The two basic design paths for an authoring aid would be:

o Man-aided Machine
o Machine-aided Man

Not all authoring-aid applications have to follow the same design paradigm. For instance, an embedded training aid could follow the first paradigm of Man-aided Machine. The expert trainer aids the machine in defining process sequences for training in tasks and defines the rules to be used by the machine in diagnosing user errors and skill levels and in programming learning sequences or correcting errors. The machine will act automatically when executing embedded training actions in conjunction with novice interactions.

The second paradigm of Machine-aided Man is more appropriate to authoring aids that aid in information searches, training plan composition, and plan evaluation and correction.

Fallouts From Current AI System Development. CAPS can exploit innovative design concepts and new technology without having to follow a high-risk development path in using unproved concepts or unreliable hardware and software concepts. The author-aiding concepts and interactive design approaches suggested by Perceptronics for CAPS have all been proven in prototype form and have been demonstrated to be usable and implementable with current commercial hardware and software components.

One of the most important fallouts of on-going AI applications work is the maturing of knowledge representation mechanisms to bridge the gap from the knowledge engineering process to the software design stage. The object-oriented programming approach has been so successful in innovative software development environments such as SMALLTALK and Symbolics' FLAVORS, that the concepts are being carried over into the development of object-oriented programming environments for structured languages like C and Pascal. The object orientation of knowledge representation approaches is also influencing the design of database management systems in terms of including object and spatial representations of data, and providing feature inheritance and class structures for objects. Although developed as mechanisms to support artificial intelligence applications, these new software approaches have shown a much broader utility in decision-aid designs that incorporate expert behaviors and have adaptability features.

Architecture Issues For Incorporating Decision-Aiding Concepts In CAPS. The approach to including authoring aids in CAPS carries the risk of being a new system design that is different from conventional practice and the attendant risk that system implementers would not know how to pursue the detailed software design and implementation. This potential problem can be offset by the specification of a structured design methodology tailored to the design of object-oriented software structures and the interactive dialog design concept. One very unique aspect of this type of design approach is that it relies on creating many generations of the system configuration by rapid prototyping, evaluation, and adaptation using inherent adaptivity features of the design paradigm.

Software Architecture. In conventional practice, software is not intended to evolve through a rapid succession of system generations but is supposed to follow steadfastly in a path of functional definition, system specification, detailed design, and implementation. Software modifications along the way are only intended to correct errors and minor oversights. The advent of software concepts to implement AI concepts created a totally new outlook on software development. The distinction between programs and data has become fuzzy and more akin to adaptable knowledge. Instead of writing new programs for each new data structure, the AI approach provided a means of creating new instances of software/data objects and adapting the features or attributes of those objects by user interaction or by rule. Obviously this software development approach is a nightmare for a conventional software configuration management control.

AI programs are not noted for their thoroughness of documentation but for their power and situational adaptivity. The most successful software development environments have provided self-documentation as the most effective means of tracking the evolution of the software structure.

Hardware. Computer hardware is always an issue for a multi-user system because cost per user can override development cost issues. New generations of computers have had such extensive reductions in cost and increases in computational power that most system architectures have incorporated distributed processing concepts and intelligent terminals even when a mainframe host remains in the system. CAPS should capitalize on the power available in the intelligent terminal for implementation of authoring aids and the tailoring of those aids to individual user skills, tasks, production roles, and preferences. The role of the host processor in CAPS is almost guaranteed by the need to provide central reference services, database administration and security for ARTEP products and source files, authority files on doctrine, production management and tasking, and other information-sharing services. High-speed local networking technology (e.g., Ethernet) can make the functions of the host or file server virtually transparent to the individual workstations.

Operating System. The choice of operating system for CAPS workstations and host functions is a decision that can make life easier or harder for the system developer and the organization charged with maintaining the system over its life cycle. In terms of software development, the choice of operating system can determine whether the system developers will have the benefit of a software development environment with tools to facilitate design, documentation, implementation, testing, and integration. Only mature, well-tested, well-documented, and well-supported operating systems with existing software development tools should be considered for CAPS.

In terms of operating system capabilities, CAPS will require networking functions, concurrent processing, support for interactive dialog (including multiple window displays), and graphics. Although these features could be added to almost any modern computer system, having these features as standard capabilities of the operating system reduces development cost and risk.

Having reliable, immediately available, and "available in the future" support to the operating system package is an essential feature of the operating system evaluation criteria. Having on-going support to the operating system ensures the continual ability to modernize and upgrade system capabilities.

Programming Language. Most operating systems that would be candidates for use in CAPS would have a broad range of available programming language capabilities. The issues of concern in the development of more advanced authoring aids are that the programming language be capable of representing an object-oriented software structure and be capable of providing the adaptability suggested for the development approach. AI programming languages can probably be eliminated from the standpoint of speed, availability of support, or level of maturity for at least the initial CAPS development. Structured languages, such as C, Pascal, or even ADA, are potential choices. FORTRAN is not a logical choice because it is ill-structured for text processing applications and has limited adaptive capability. In any case, the language selected should be supported by a software development support environment and should have some level of self-documentation to support the evolution of system capabilities.

Man-Machine Interface Requirements

We believe that the design of the CAPS user interface must exploit the technology advances in intelligent workstation hardware, sound human factors principles, and a more appropriate paradigm for structuring the interactive dialog on the basis of a cognitive model of author problem-solving processes. The approach to the design of the user interface exploits the ability of state-of-the-art workstations to provide multiple window displays, graphical as well as alpha-numeric presentations of information, and provide local storage and computation to tailor the interactive dialog to specific users.

Although no new technologies are represented in the CAPS design approach, a new level of interface functionality has been introduced into the system architecture. That new level provides the mapping of the ARTEP author's conceptual models of the domain and ARTEP production processes to Automatic Data Processing (ADP) functions. Most conventional interface designs do not provide this level of mapping and require the user to deal directly with ADP representations of information (e.g., files, records, queries, commands). From the results of work Perceptronics has performed in the areas of cognitive modeling of analytical processes, it has become apparent that the problem-solving processes revolve around complex conceptual models that carry domain knowledge, specific knowledge of situations and tasks, and procedural knowledge.

The implications of this research and experiments in adaptive user interface designs have clear implications for the structuring of man-machine dialog. Successes of systems like the Apple MacIntosh that use iconic representations of system functions and data are also evidence to support the need for the additional level of interface design that addresses conceptual models of processes and information.

Unlike the MacIntosh, which is a general purpose computer system, CAPS must tailor its interface design to the ARTEP production domain and to the specific skills and knowledge of authors and support personnel that constitute the user population. The desktop metaphor used in the MacIntosh to represent "conceptual models" of ADP functions is not sufficient for CAPS although it works well for novice users doing word processing and management information tasks. For CAPS, the interactive dialog must consider that icons or dialog semantics must relate to concepts that carry knowledge about tactical missions, tactical units, tactical environment, operations, force structures, etc.

As a general rule, the CAPS user will not be an ADP expert. A major thrust of the CAPS interface design approach should be to make the interface understandable and usable by the novice.

Perceptronics has formulated a series of general guidelines for implementing the proposed man-machine interface for CAPS. These guidelines are organized into three distinct categories.

- o Features that address user ADP skill level,
- o Features that support the user's cognitive activity,
- o Features that give the system credibility.

These guidelines are summarized below:

Interface Features That Address User ADP Skill Level. The following features focus on those attributes that allow the system to adjust or be adjusted to the skill that the user has with data processing systems.

Thinking in Domain Terms. Users will tend to think in terms that are relevant to the domain and the tasks they are assigned, rather than in ADP terms. Data structures used in the interactive dialog should relate directly to the domain and have a logical structure in terms of the domain and authoring tasks. In some cases, this may mean that the internal system representation of data may have to be reorganized or tailored for specific user processes.

Understanding What the System Is Doing. A novice user may have no idea what the system is doing or may have an incorrect concept of what the system is doing and what is expected from the user as input. User interface features that can deal with this issue include:

- o Providing examples of user entries at required data entry points,
- o Providing concrete models of the database structures,
- o Providing feedback to the user on how the system is interpreting a query or other user commands.

Language of Interaction. Terminology used in formal logic or that is understandable by a programmer may be a puzzlement to a novice user. Potential solutions to this problem include:

- o Using domain terms rather than programming terminology,
- o Using menu-driven formulation of complex data entries,
- o Using query by example,
- o When using formulas is necessary, providing diagnosis and error recovery for expected types of errors.

Help In Using the System. All users will periodically need help in using the system, some more frequently than others. The interface design can address this problem by providing:

- o Simultaneous display of help with the user's problem,
- o Help information should be specific as to context of process/step,
- o Guidance and immediate error correction for processes,
- o Clarification dialogs for more ambiguous situations,
- o Display of understandable error messages with indications for corrective measures,
- o Avoidance of semantically confusable names or abbreviations,
- o If optimum navigational strategies exist in the system, the user must be made aware of them (using mechanisms such as visual representation of paths/steps),
- o Automatic presentation and cueing to options.

Time Invested In Learning the System. Novice users may not have the time or the desire to spend up-front effort in learning the system unless the organization is willing to devote time to formal training classes. The optimum solution is for the system to provide embedded training and on-line tutorials.

Ability and Willingness to Type. CAPS users may come from disciplines and assignments where typing and ADP skills are neither required nor used. As such, typing skills may be a problem for an interface design that is oriented entirely to keyboard data entry. Potential solutions include:

- o Direct pointing (touch screen, mouse, trackball, or other cursor movement devices) to control interaction or designate data entries,
- o Menu selection whenever possible,
- o Provision of default data entry or steps,
- o Icon manipulation,
- o Shorthand data entry formats.

Ability to Make Precise and Complete Specifications. A novice user may be unfamiliar with the precision and terminology requirements required for the execution of ADP functions. Potential solutions to this issue include:

- o Use of forms, menus, and templates for controlling data entry operations,
- o Use of data dictionaries or authority files to cross reference from imprecise to precise terminology,
- o Use of spelling co‹rections on command names and parameter selections along with interactive feedback.

Fatigue and Boredom. Long sessions at user workstations and time pressures of product production may create fatigue problems. Similarly, extensive regularity and labor-intensive data entry processes may compound the situation with boredom factors. Potential solutions to these issues include:

- o Use of templates and process monitoring to aid in monitoring user vigilance,
- o Embedding of alarms or alerts to vigilance failures,
- o Reduction of labor-intensive data entry operations with automated aids and preprogrammed functions,
- o Cross-training and increased variety in authoring tasks.

Goal Orientation. The system designer must recognize that the user is goal-oriented and that the design job does not end at the point of providing general ADP functions. Especially for the novice, it is important that access to ADP functions be organized in terms of how the functions would be used in achieving the user's goal. Interactive designs that depend solely on the user to supply goal context and to adapt the system functions to that context are derelict in their support for the military user. Modern system designers should be able to support the goal context of the user process by providing:

- o Cues and supports to current goal and task context (especially in those cases where many production goals exist concurrently),
- o Hierarchical structure of goals, objectives, and tasks to provide the user with a reminder of the current objective to the big picture.

Concurrent Processes. Authoring functions, as with many other problem-solving tasks, requires the user to be able to do more than one thing at a time on the basis of time pressures, priorities, dependence on external inputs, and availability of information. New hardware and software technology available for the CAPS workstation can support features that aid in this aspect, including:

- Providing discrete windows for different asynchronous tasks, access to help information or reference data, or providing alternative displays of data,
- Providing process-context sensitive help information and cues so that the user does not forget what the current objective is,
- Support for multi-task processing so that users can be productive while waiting for return of requested information, waiting for printing of documents, or when relevant information must be overlaid on the display to support a current process.

Interface Features That Support the User's Cognitive Authority. If the CAPS system is to have the objective of aiding the problem-solving activities in the authoring process, then some of the burden of the interactive dialog must be assumed by the computer side of the dialog-control mechanism. To accomplish this objective, the system must have features in addition to those that address needs of the novice user. Most importantly in this aspect of interactive dialog design is the ability to map the functions of the interface and the database to the goals, knowledge, and goal-oriented processes of the user. Most computer-aiding systems have not provided this kind of dialog support because user interactions have been homogenized into context-independent ADP functions.

In those cases where the machine is organized around goal-oriented processes, the interaction sequences are inflexible and stilted. Cumbersome menu operations, data entry steps, and inflexible data entry steps are not adaptive to the nature of knowledge-intensive problem-solving processes that typify a CAPS authoring task or many other types of military analysis. Guidelines for the user interface design that promote the support of the user's cognitive process are as follows:

Mixed-Initiative. This means that the system shares the initiative for the control of the interactive dialog with the user. The system can initiate dialog by volunteering information, providing feedback, controlling display formats, issuing alerts or status information, diagnosing errors and corrective actions, or taking default actions in the context of specific tasks. The user controls the dialog by navigating between goal and process contexts, executing task or step requirements, overriding system-suggested actions, and controlling rule systems that execute automatic actions. The mixed-initiative dialog approach is inherently flexible in its response to goal-oriented process requirements.

Vigilance Support. Given a model of the user goal structure, the system can follow the progress of task completion and compare it to performance models for the task or to past task executions.

Navigational Support. Accomplishing necessary actions in a goal-oriented process requires navigating through the hierarchy of objectives, tasks, steps, and supporting actions. A navigational aid would keep track of current goal context, actions completed, actions required, and would provide a prescriptive path through those actions and processes. Navigational aids can also aid in the exploitation of relevant background knowledge (past products, process sequences, references) without losing track of the current objective or processing state.

Display Surface. The user should have control over the format of display surfaces in an effort to take into account the goals, problem-solving style, priorities, and information needs of current tasks. Multiple window display capability, stored process sequences, and window management functions make it possible to tailor display surfaces to individual needs.

Control Surface. The user may need to create command sequences and store these for later use. These sequences can be initiated by the user or by the system upon reaching a prescribed process context. This mapping of system functions to higher level actions provides for an adaptive and evolving interactive dialog capability.

Consistency. For any computer-aided system with a large repertoire of functions, it is difficult to maintain consistency in how various similar functions are performed. Whether CAPS eventually leans toward a pointing metaphor using icons, forms, or menus as its principal structure, there should be a consistency across all system functions and a consistency with domain concepts.

Short-Term Memory Support. Human short-term memory is limited. To support the user's short-term memory, the system may need to be able to store and retrieve scratch pads that could be used for transfer of data between tasks or windows, maintain a personalized database, set up reminders, create customized alarms, or customize help functions.

Context Maintenance. The context in which the user is working should be monitored and preserved. Users may need to restore a context following an interruption. Context maintenance is needed for the system to be able to select help and window configurations preset for specific tasks.

Context Customization. Processing models may be available for users to copy or modify into new process sequences (e.g., the development of collective task requirements in CAPS). As special ARTEP or training plan requirements emerge, standard sequences may be customized into special versions.

Interface Features for System Credibility. Any system that hopes to provide automated aids to problem-solving processes must have a means of establishing and representing its credibility to the user. The following guidelines are seen as relevant for implementing author-aiding functions in CAPS.

Audit Trailing and Reasoning Chains. Although the user may not wish to audit every automated action of CAPS, there must be some means to verify what the automated process has done and to backtrack and fix actions if required. The CAPS interface design approach aids in keeping meaningful audit trail information by structuring user and system processes into specific objective-oriented processes. Navigational aids can also provide the data structure for retaining audit trails and other information needed for intermediate protection of process actions.

User Control. ARTEP authoring is a knowledge-based problem-solving process that utilizes varying and unpredictable levels of background knowledge as well as external information sources and situation-specific data. A CAPS interactive dialog controller could never hope to predict and control the multitude of process sequences that could occur in the authoring process. For this reason, the user must be able to maintain control over the overall ARTEP production sequences and tailor those processes as required to particular training product requirements.

The cognitive model approach provides a dialog structure that is focused on the problem-solving process itself and not on the substance required to perform the problem-solving. The user maintains control over the substance of the problem-solving process by investigating available information sources at the time the process is performed and depends on his own knowledge as a function of experience, confidence level, and time pressures of producing specific products. As the system-automated processes acquire credibility, the user can depend more and more on automated aids for greater product quality under the time pressures that prevent the use of all available information sources and thorough evaluation of product requirements and constraints.

Maintainability and Extensibility. The man-machine dialog should be documented and implemented in such a way that dialog errors can be identified and remedied and improvement made possible. In addition, the means of implementing the dialog must be sufficiently flexible to support adding new tasks without disrupting existing applications. This could be accomplished in a building block fashion, maximizing the use of existing facilities to create a new task context. It is important that the user not require expertise in computer programming to be able to conceptualize alternatives in the dialog with the system.

Error Recognition. Error recognition can only be accomplished to the extent that a performance model exists. As more of the user's goal structure and process performance can be represented in the system, then more diagnosis capacity can be built into the system. The CAPS design approach extends this ability by addressing the overall goal structure of the ARTEP production process.

Dialog Control. Based on the the previously described guidelines for the CAPS user-machine interface, the highly graphic interface concepts developed by Alan Kay and his colleagues at the Xerox Palo Alto Research Center (PARC) are the most appropriate techniques for the CAPS interface. Although PARC technology includes a number of innovative concepts, including "windows," "pull-down menus," "mouse cursor control," etc., the central concepts of the PARC interface technology are 1) concepts represented as "things", shown graphically as "icons," and 2) the "desktop analogy," in which the computer screen is considered to be similar to the user's desktop.

Icon representation of concepts is a powerful tool for interface design, since it capitalizes on visual cognitive processing. A vast majority of human cognitive processing is based on information acquired through the visual system, whereas a smaller percentage of cognitive processing is based on acoustic, or language-based, information. Thus, we would expect better (e.g., faster, more accurate, less fatiguing, etc.) performance with a computer system in which the majority of information is transmitted to/from the user via graphic representations of data, files, procedures, relationships, etc.

The desktop analogy builds on the icon representation of "concepts as things," to provide a graphic representation of the contents of the computer system in a manner similar to folders and papers on an office desktop. Thus, the user can move icons representing pages (e.g., text pages, drawings, or spreadsheet pages), which look graphically like pieces of paper, and then place them inside of icons that look like file folders. Folders can also be placed in other folders, or in icons that look graphically like filing cabinets. Similarly, the user can arrange things on the workstation screen much as he/she would arrange things (papers, folders, scissors, tape, etc.) on a desk.

Although the graphic representation of concepts as icons is powerful, in and of itself, the central core of the technology arises from the fact that the user can move and arrange the icons in space (on the screen). Thus, we capitalize on the uniquely powerful human capability of spatial data management. Recall the proverbial office colleague who always has an extremely disorganized desk and office, but who is able to find anything within just a few moments because he knows "just where he last put it." In a similar manner, the desktop analogy for a computer screen allows the user to arrange icons in whatever manner he/she choses, and to then retrieve whatever is desired, based on the spatial arrangement that he/she created.

Other features of PARC technology then evolve naturally from these two basic concepts. To be able to place graphic icons easily on the screen, a "natural" and easy-to-use method of "pointing" is required. A number of devices are potentially available (e.g., mouse, trackball, joystick, light pen, cursor keys, etc.) and each device has an appropriate application, depending on the demands of the specific task. In the case of text and graphic processing, when the user is seated at a desk with a keyboard for primary text input, and the screen arranged nearly vertically in front, a mouse or trackball is most appropriate.

Given the requirement for a mixed-initiative, menu-driven system, and the existence of a handy "pointing" device, pull-down menus have proven to be an effective manner of choosing among many alternatives. A menu bar that is always displayed on the screen (usually at the top of the screen) shows the major categories of choices that are available at any time. The choices on he menu bar can be shown with icons, with words, or both. The user then "points" (i.e., moves the cursor) to the desired choice on the menu bar, which causes the full menu for that choice to appear (typically as a list that appears as a window that unfolds below the chosen item on the menu bar).

In keeping with the desktop analogy, a number of concurrent processes can be active (i.e. "open") at one time, similar to the many different pieces of paper that an office worker may have scattered about on the desk. Each of these open processes can be shown as a "window," and like the pieces of paper on the desk, windows can overlap other windows, or can exist side-by-side. In the case of overlapping windows, the window on top is the one in which the user is currently working. In the case of side-by-side windows, the user could be working in both windows, perhaps moving text or graphics from one to another. This latter instance would be quite typical in the ARTEP production process, since many sections of ARTEP documents are taken, verbatim, from other documents.

Text Processing. There are two main types of data that an author uses during the MTP production. The graphic data are described below. Here we will concentrate on the textual information that comprises about half of a typical training plan.

The computer-aided ARTEP production system must be capable of supporting the usual text processing capabilities, including 1) word wrapping, 2) deleting, moving, and inserting characters, words, sentences, and paragraphs; 3) inserting, deleting, and changing rulers (line, paragraph, or page formatters); 4) changing fonts styles and sizes; 5) saving and retrieving documents; etc. However, the CAPS word processing capabilities must also include some capabilities that may not be part of many typical word processors. Because many of the ARTEP documents are quite long, an easy-to-use means must be included for moving rapidly (i.e., within 1-2 seconds) anywhere within any document that is currently "open." Also, since many documents are created by combining entire sections from other documents, the system must support readily and easily the functions of "cutting & pasting" large sections from one document to another. Indeed, such "cutting & pasting" should be performed automatically, based on the relationships among ARTEP documents that will be built into the database management system.

Because text and graphics are so intimately related in ARTEP materials, the text processing capabilities in CAPS must also be integrated with the graphics processing capabilities. Thus, the two functions must operate simultaneously, and all similar functions (e.g., text editing, cutting & pasting, moving "objects," etc.) must be performed using identical functions.

Fortunately there is a large amount of repetitive information in the contents of a training plan. Thus a large part of the effort of supplying the appropriate language will come first from the loading of the database from the present hard copy and floppy disk formats. This information itself will be in a relational form, hence we wish to incorporate it in many places, but in a boiler plate form. Thus, when we make changes to one copy, the system can inform the authors of all places where the data are used, to see all the places where it is potentially required to review the intended change. This is a natural result of the relational database format.

The actual editing itself :an be handled using text formatters similar to those incorporated in products such as Microsoft Word or Apple MacWrite. This class of product will permit a ·atural mapping of the text process to the graphics screen interface, which is needed since we are producing ultimately hard copy paper output that needs to be carefully "pasted up" to be readable.

The most significant editing of the data will be during the initial loading phase. It is expected that there are major amounts of inconsistency among the terminology used by different authors, reference names have been changed, etc. The solution to the problem will be the development of a standard set of relational queries hat will gather together standard terminology, showing unique elements, and allowing the data administrators to select among various alternatives and replace globally for the appropriate choice. This set of tools will also be used as a consistency check on the final and intermediate forms of each document, to ensure editorial consistency. A certain amount of this can be done while entering changes, but bulk reviews will have to be periodically scheduled.

Graphics Functions. The entire graphics subsystem is involved in the production and incorporation of line drawings. Approximately half of the data are in this form, hence it is essential that this process be simple, and that there be sharing of information among distinct drawings. Hence we need an object oriented drawing system, in which objects are composed hierarchically of more basic elements. These basic elements are graphic objects such as rectangle, ovals, vertical ad horizontal lines, etc. As well, we must be able to name and reference (via the relational database) subdrawings, so that we can include a standard image (icon) of a tank whenever it is needed, without having to draw it each time, and also giving the capability of making a local change to the standard icon, and having it propagate to other places. Again, when such a change is made, it is left up to the author to verify whether he wants to change every place where this icon is referenced, or to keep some of the references to the original object intact.

An excellent model of an application that non-draftsmen system users are capable of using is Apple MacDraw (or LisaDraw). In this application, users select among a handful of general tools, which can draw lines, boxes, curves, etc. By selecting objects, individually or in groups, they can be moved across the screen. Objects have well defined "handles," or special points, which can be used to stretch them in one particular axis if desired. As well, these handles can be used to align a series of objects to each other, say centered vertically, etc. This application is capable of expressing all the objects we have seen in typical current ARTEPs, and no further creation is needed to handle the objects in question.

The last piece of software that is needed, but which needs to be developed is what we refer to as the "page layup" function. In this phase, actual production consideration of the hardcopy manuals must be considered. Page size, and borders must be dealt with, and the illustrations shrunk or expanced to fit. Similar juggling with text, often requiring minor violations of margins etc, will be done on the screen of the workstations (hence the desire for quite high resolution devices), prior to transfering to output mechanisms.

There are four levels of graphics output required, depending on the quality and permanence needed. One may see the formatted page on the screen, and perform the manual adjustment needed before committing the output. Next one can produce "draft quality" output on an inexpensive dot matrix printer locally connected to each of the workstations. This will be used for keeping file copies of working versions, etc. Once these are satisfactory, the page layups are next sent to the centrally served laser printers. These devices are much slower to access than the local devices, since there is some format conversion needed to supply the graphics language needed at the laser printers. Once these masters are acceptable, another option will be to produce output on professional phototypesetters. Such a transfer can be done via magnetic tape. The format of the information needed for the phototypesetters is the same as we expect to be needed for the central site laser printers.

Database Management Requirements

General Goals. The underlying software required for CAPS falls into several areas, as follows:

1. A Database Management System (DBMS) system for storage and retrieval of Mission Training Plan data .

2. Documentation retrieval and tracking system used to manage internal distribution, checkoffs, and routing.

3. Graphics editing and presentation subsystem.

4. Text processing tools, used for text editing, formatting, spell checking, etc.

5. Version control methodology.

Choice of Relational vs. Hierarchical DBMS. A superficial examination of the existing hardcopy MTPs might lead one to the conclusion that a database system with a strong orientation towards tree structured data closely models the underlying information. Such a hierarchical system would contain the internal links and very carefully organized series of nested tree structures that are evident as the dominant means by which previous authors have chosen to present the MTPs in paper form. (Of course, this would not apply to the significant number of crosswalks, matrices, indices, and other material that appear to be auxilliary in nature.)

In fact, this observation probably does reflect the state-of-the-art of the current linear hardcopy presentation system, and is perhaps accountable for a significant number of the problems associated with the production of MTPs. Organizing data in this form provides for arbitrarily complex internal references and cross-references, but all the linkages are generally provided in an explicit form, rather than implicitly via the data references themselves. Hence, it becomes nearly impossible to provide a means to do large, global changes to referenced information and find all affected areas, without literally performing textual searches. But even such searches are quite limited, due to inconsistencies in terminology, throughout such a database.

The analysis of the existing training products in the second chapter has attempted to indicate that, behind the current MTPs, is in fact a highly structured conceptual database, with significant amounts of internal parallelism and relationships formed by linking various exercises one to another, etc. This analysis leads us to look towards a symmetrically structured form of data representation (e.g., a relational database) as providing more value. In particular, we view the current presentation (including delivery vehicles, e.g., paper) of an MTP to present only one very specific means of understanding or studying the actual data itself. They model the views of the data that authors traditionally use and provide. But they do not reflect either the conceptual database nor the computer implementation.

Assists, such as the STX-to-FTX matrices, are examples of derived views of the database that have historically proved useful to trainers and MTP authors. We believe that users of the CAPS system will rapidly discover other fruitful ways to look at the information contained within the database.

Views of the Underlying Data. A view of the underlying structure has been presented as being a multi-dimensional "cube". The general relationship among the elements within that cube have also been shown. However, the DBMS forces a presentation of the data in the form of "tables" (e.g. a two-dimensional structure). Is this somehow missing the very point to be made?

Indeed, an implementation requires some two-dimensional view of the data, but this should be understood to relate more to the fact that the data have been indexed along certain preferred axes from the perspective of computer performance. An appropriate relational DBMS must provide multiple "views" of the data (e.g., despite the fact that internal to the computer, data are projected along certain axes, queries and other tabulations can be fully symmetrically applied). Through careful use of multiple indexing, some of the other search and selection axes can be reasonable performers.

Another advantage of the projected and tabular representation is that a large fraction of the total number of cellular intersections in the cubic view are null. Hence, this format will not be used in managing the data.

Nature of the Authoring Environment. A form of multiuser system is a requirement for CAPS. There is not any current need for the kind of controls typically found in financial accounting systems, for example. Different individuals are not modifying different fields in the same record, or making changes that must be synchronized in realtime, for fear of providing an inconsistent state. However, as the proposed CAPS database implementation will telescope data from different echelons, and from what today are thought of as quite different databases together, it will be necessary to lock rows in a relation during their update. A system that locks a whole relation will not be acceptable.

CAPS itself is a form of tool. And the development (as opposed to deployment of CAPS) is heavily dependent on a host of very flexible and advance tools, used in an interactive environment, exploring rather new areas of the data. Hence, the most productive means to develop CAPS will be in environments that emphasize this feature.

Either of these two operating systems can provide for (1) the limited number of users needed, (2) connection of the users through a global or local area network, and (3) the distribution of documents assisted by electronics mail and other distribution systems. These operating systems also are very strong in the area of text production tool, which is required for the linear hardcopy MTP output.

Graphics requirements. Graphics is needed in several distinct forms within the CAPS systems. These can be viewed in three main groups:

1. Drawings that are part of the MTP being produced,

2. Graphics used to show structure in an MTP (e.g. charts displaying the relationships among FTX and STXs),

3. User interaction graphics, for manipulating high complex non-linear data on a computer screen.

The storage size required for graphics of type (1) can be derived, since that type is, by far, the most predominate in MTP documents. First, the basic nature of the encoding used for drawings must be determined.

An overview of the contents of present MTPs shows that line drawings are considered acceptable. An object-oriented image description system lends itself best to the style of drawings used. Drawings kept in this form can be scaled for display on a video screen, and then later printed out in higher resolution on a laser printer for actual distribution. This is not the case for bit-mapped graphics.

An object-oriented system, coupled with a relational DBMS, can be used to nest object descriptions together. This will provide for ensuring global consistency in the iconic representation of objects, and for uniformly updating changes in the future.

Object-oriented systems will also provide for much lower storage requirements than a bit-mapped representation. Initial estimates indicate that one MTP drawing requires no more than 500 nodes/drawing. Each node might be a 8-tuple of 4-byte parameters. Such a drawing would need 16 Kbytes of storage, yet could represent excellent quality drawings. Given the relational nature assumed for drawings, perhaps 2,500 drawings per MTP (all echelons) may be required. This would require 40 megabytes of disk storage.

Text storage size. The analysis of existing MTPs revealed a page count of about 5,000 pages, with about half of the pages covered with figures or tables. Allowing for storage of multiple versions, and typical internal storage fragmentation, we estimate that 50 to 100 megabytes of storage are needed.

Later in the design, once the actual table layouts and field names are specified, the size can be re-estimated more precisely. For instance, it is expected that the Task Documentation Summary Table is the larger table, but has at most a number of rows in the low thousands.

Number of files, relations, and storage packing. It is evident from the sample analyses presented previously, that queries and displays will incorporate data spread across a large number of relations, perhaps in the dozens. The exact number may in fact be much larger if we were to choose additional structuring based on echelon, although this will not be done except in case of performance problems. Any selected system must provide for a large number of relations open at once.

The size of the records must accommodate highly variable quantities. Some fields may be bounded by fairly small sizes (e.g., textual description fields which may be under 100 characters long). But the system must also accommodate fairly lengthy text appendage, which can be thousands of characters long. Graphics images will also be stored in such formats. (The analysis above indicated that storage will certainly exceed 25 characters/bytes, but not exceed 64,000 bytes.)

Given the evident variability in storage requirements, plus the fact that large number of fields (even fixed size fields) may be totally omitted, the DBMS must pack such data into the minimum possible file space, plus a small overhead quantity.

DBMS front end, language interfaces, and parsers. It is evident that the DBMS must be hidden within a user interface designed specially for the CAPS system. As as example of the actions that the support system would be expected to provide, would be to extract what the DBMS considers variable length text strings but display them to the user as a structured graphics image. The user must then edit the image, and it should be stored back. The DBMS is not required to provide the graphics editor. Similarly, in the area of the display and editing of free-form English, other software will carry out the needed manipulations.

Significant use of pseudo-English parsers will be made during both the development phase of CAPS and during production. Initially, such parsers are needed to support the exploratory and development usage of the system. It can also provide a flexible, interactive "what-if" facility for permitting MTP authors to do additional analysis and view data in ways that were not anticipated in the canned CAPS view and matrices. These features will significantly enhance the quality of the MTPs developed with the system.

The elements required are:

a) CALL-level linking to database primitives
b) Macro preprocessor conversion to HLL
c) Run time call with unparsed request string
d) Pseudo-English parser with "what-if" screen reporting

The interfaces from languages such as C, FORTRAN, Assembler, and LISP may also be requirements for CAPS.

Computer System Approach (Hardware and Software)

This section brings together the requirements and relevant technologies from the previous sections, and identifies a number of possible vehicles for target development and deployment. It attempts to reduce the design space to a few highly qualified choices, and to bound the concomittant hardware and software acquisition costs. It does not provide guidance regarding the sizing or costs of CAPS-specific development.

The Multiuser, Interactive Environment. Chapter 2 has identified the number of co-authors for each MTP to be in the region of a dozen, each with distinct functional responsibilities. They all are physically near each other (on one post), thus, it is feasible to consider them linked together via high bandwidth (e.g., multi-megabit per second) networking.

The environment selected is optimized for the team level, and each separate development will have its own copy of the system. Typical variances to be allowed for in site-to-site changes are in the size of the database, frequency of modifications, and a small variability in the number of authors. Exchange of information between sites will be done by tape exchange, although this must be supplemented by a wide area network with a lower speed interchange mechanism.

The actual deployed systems are not significantly different from the development environment itself. The users of the system will share the same set of tools and related tasks that the initial system implementors do. For example, the DBMS itself can be used for project tracking during development, Similarly, software for version control, text and graphics processing for system documentation, etc., can be be shared for development and deployment. Hence we will select one system that will accomplish both goals, economically.

The system does not require a massive amount of computing power. The performance of the DBMS system need only be average, since it is used to build fairly static tables, answer queries on an infrequent basis, etc. The system will mostly be taxed by the graphics and AI requirements.

Peripherals. The disk storage requirements are estimated to be in the 100 to 150 megabyte range. This does not tax any of the above proposed configurations.

A suitable backup system for data must be incorporated. It must also provide for industry standard exchange of data among CAPS sites, and should also provide for import of distributed software from vendors. A 9 track, IBM compatible streamer tape approach is appropriate.

A high quality, medium performance laser printer system is needed. It will be used as the primarily output generator on the system. It should be supported by (optional) local, lower performance printers, which are physically close to the authors.

Workstations Employed. Standard ASCII textual terminal are not suitable for the integrated production environment needed for CAPS. The class of graphically oriented workstations that meet the user interface and graphics display requirements are machines in the class of Apollo, SUN, Xerox STAR, AT&T 7300, high resolution versions of IBM PC/AT. The chosen machines must also possess a very high speed host attach rate, for graphics storage and retrieval.

Micro-networks vs. Minicomputer Host. The high performance graphics requirements have implied the usage of workstations with considerable computing power and interconnect bandwidth. The amount of computer power needed for additional tasks, in fact, is quite limited, and only occasionally called on. This leads to the design of a system based on having the primary computing activity in the workstations themselves, rather than in a central host. This will provide a system that can be scaled up and down effectively, both in the number of workstations and in their size. Clearly, a software environment is also needed that is available across a wide range of performance points, from micro to mainframe. (This requirement rules out an otherwise interesting alternative architecture, namely the use of a database machine, such as the Britton-Lee engine, for the storage subsystem. The advantages provided are minimal in return for the narrowing of design points for additional systems in the future.)

This tends to rule in favor of UNIX as the host operating system environment, since it has been ported to such a disparate set of different hardware environments. A consistent set of tools across all machines is desired, hence a UNIX-based server environment is required.

A network capable of spanning a base, running at several megabits per second is required. An Ethernet system running over broadband, or an ARCNET baseband system would be acceptable.

Database Software. There are several UNIX-based database systems that generally meet the requirements described in Section 3.4.2. (Among them are Oracle and Ingres.) However a final selection requires more detailed design, and then detailed matching of requirements against these products.

The database software must run on a series of machines, from micro to mini based, in a fully compatible fashion. Oracle seems to have no identified conflicts with the system as envisioned. It does support unstructured data of up to 64,000 bytes in a field for example, and packs data very well. IIt provides for very flexible view management as well.

Prototypes and Growth. The system as described is capable of up-scaling in terms of the number of stations, power and class of workstation, amount of disk storage printers, etc. without any changes to system software or organization. Hence, the system can be prototyped and do the development with fewer than a complete set of workstation nodes, and prove the concept on a small scale.

CAPS SYSTEM DESIGN

In this chapter we describe the design features to be included in the CAPS system, including the data input system, the manner of representing the ARTEP database to the user, the database management system, and the information processing (computer) selection.

Data Input

Figure 30 shows the major components of the CAPS User Input System. The primary function of each component is described in the following paragraphs.

ARTEP Database. This is the primary information that is stored by the system. All of the other functions are for the purpose of modifying this primary database. The structure of the ARTEP database is described below. The database consists of a number of different files. The records in a file are the individual training documents and include the following:

a. Mission Files for each echelon where records are the specific mission descriptions:

- o squad missions
- o platoon missions
- o company missions
- o battalion missions

b. Soldier's Manual File for each MOS where records are the *descriptions for the individual soldier tasks.*

c. Military Qualification Skills File for each echelon where records are the specific qualifications for a given leader, staff or command:

- o MQS 1 for Squad Leaders
- o MQS 2 for Platoon Leaders
- o MQS 3 for Company Command
- o Doctrine Manuals for Command and Staff

d. T&EOs File for each echelon where records are the specific T&EOs:

- o Squad Collective task T&EOs
- o Platoon Collective task T&EOs

e. Drill File for each echelon where records are the specific drills:

- o Squad Collective Task Drills
- o Platoon Collective Task T&EOs

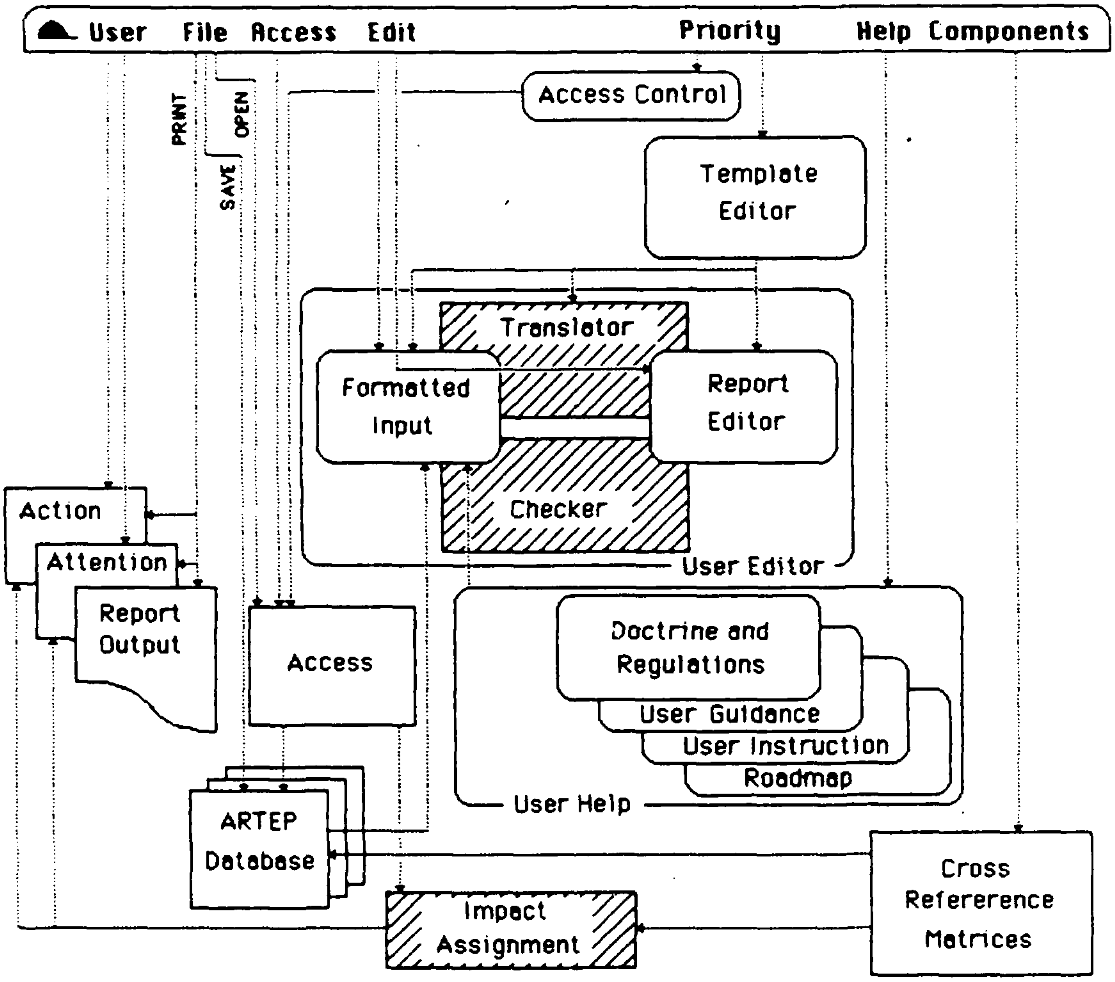


FIGURE 30. CAPS User Input System

f. Situational Training Exercises (STX) File for each echelon:

o Squad Mission STXs
o Platoon Mission STXs

g. Field Training Exercises (FTX) File for each echelon:

o Company Mission FTXs
o Battalion Mission FTXs

User Editor. The User Editor is the primary component for making changes in the ARTEP data base. It consists of four components. The user interacts directly with the FORMATTED INPUT component or the REPORT EDITOR component. The TRANSLATOR and CHECKER components are transparent to the user but interact with the user input as described below.

Formatted Input. This component consists of a template or form that requests specific information from the user. The nature of the information requested depends on the type of document under consideration. The template for each type of document will differ. The intent of this form is to facilitate user interface with the system. The system combines user-supplied information with the system-supplied information to create the report format that the training document will have in the field.

Translator. The translator takes the data from the template formatted input and converts it to the report format appropriate for the specific training document or mission statement. The report format is the format that the training documents will have when used in the field. The system adds the information which it can supply to the information supplied by the user. The user inputs only the essential new information to form the particular training document under consideration.

Checker. The Checker is the intelligence in the input system. The checker monitors user input to see that it is within reasonable limits. It also indicates when the user uses words and terms that are not consistent with the doctrine and regulations for a given training document.

Report Editor. The Report Editor allows the user to edit the translated document. It is possible that the translator will introduce awkward phrasing, inappropriate sections, etc., because of the limitations of anticipating the many needs of a particular training exercise. The Report Editor allows the user to re-phrase given items to make them more appropriate to the end consumer of the training material. These user modifications will also be subject to the checker, to the extent possible, to determine if the user makes inappropriate changes. The checker will restrict changes in format that are inconsistent with the regulations or doctrine currently in the system.

User Help. Five types of help are available to the user while using the CAPS system. One, ABOUT CAPS, is available any time. The other four (DOCTRINE/REGULATIONS, GUIDANCE, INSTRUCTION, and ROADMAP) are specific to a given input template and are available whenever the user has a specific input template active in the Input Editor.

About CAPS. The ABOUT CAPS option provides the user with information about the system and system commands. A sub-menu enables the user to find the necessary system help quickly and pertaining to a particular function.

Doctrine/Regulations. The Doctrine/regulations option provides the user with the doctrine(s) and/or regulation(s) that are pertinent to the item being requested by the template.

Guidance. User guidance is a form of hyper text that allows the user to obtain definition(s) of the terms involved.

Instruction. User instruction is a mini tutorial concerning the document under consideration. This facility is aimed particularly at first time and naive users who may not be familiar with the process recommended for developing the training document under consideration. Selecting INSTRUCTION will provide a presentation complete with rules and examples for how to prepare the document under consideration.

Roadmap. The ROADMAP option is a graphic representation of the underlying CAPS database that gives the user a "picture" of the document(s) that he is modifying in relation to the whole system. The document(s) currently being updated by the user are shown in highlighted form.

Cross Reference Matrices. The cross reference matrices and associated components are the heart of the interactive CAPS system. The relationships among the ARTEP databases and training documents are contained in these matrices, providing the basis for moving automatically from one database or document to another. The cross reference matrices also provide a conceptual basis for graphically illustrating the current state of the CAPS system to the user. The structure of these matrices are described below (CAPS Data Representation).

Impact Assignment. This system component is transparent to the user and is used to assign attention and action items to appropriate users. When a user is registered, his identification indicates the levels of training documents that are within his/her domain and the specific training documents for which responsibility has been assigned. This information is used by the impact assignment function to send the impact of changes in the cross reference matrices to each of the users affected by the changes.

The cross reference matrices are the source of "down stream" action items and "up stream" attention items. Adding or deleting a training document from the system has impact on all of the component documents that are included or cross referenced. Adding, deleting, or changing a document has impact throughout the system. The cross reference component sends the appropriate attention or action messages to the other system users who are impacted by the change.

Attention and Action Reports. These reports are available to a user whenever he/she logs onto the system. The Action List is a menu of the particular documents that a given user needs to change to make them consistent with changes made "up stream". This list reminds the user at a particular level with particular assignments which things that they need to change. The Action

List is literally the list of assignments that have been generated by the system as a result of changes made "up stream" from the particular user. Different Action Lists are appropriate for different authors. An action report indicates that a change has been made in a given collective task, mission, or training document that requires a parallel change to be made in other documents that are within this particular user's domain. The user can observe the changes that have been made by calling up the appropriate documents and can then make the necessary changes in the documents thus affected.

The Attention List is a menu of suggestions about changes that may be desirable because of insights, inconsistencies, and other changes made "down-stream" from the current user. The Attention lists requests that the current user make necessary decisions about changes that are suggested but does not mandate these changes, as does the ACTION LIST. Attention items come from "down stream" and are suggestions to a higher level user that there may be an inconsistency at a lower level that would suggest a needed change in a mission statement or other higher level training document. Because the user receiving the attention item is at a higher level, he/she is in a position to decide what changes are necessary to overcome the noticed inconsistency. Such changes may be changes in a higher level document or may be action items for the originating source of the information to change the lower level document, thus resulting in an action item "down stream."

Template Editor. This is a special input function that is used by a priority user whenever there is a change in doctrine or regulation that affects the format of any of the documents in the system. This function allows the user to change the templates which govern training report generation. This same function also allows adjustments in the intelligence of the system; e.g., those items that are used to judge the appropriateness of a given user's input. The Template Editor enables the system to be easily modified to accommodate anticipated and unanticipated changes in the way training is conducted. This component enables the CAPS system to always be current.

Access Control. Because CAPS is a multi-user system in which different users have different assignments and priorities, it is necessary to have control of access for the different users. This is accomplished by several of the components working together. The first is a registration system where a Priority User (Super User) assigns a given user a priority level and a domain of specific training documents. This information is fed into the ACCESS component that controls access to the database. Note that information concerning access is also necessary in assigning Action and Attention reports.

The Access Control also allows the Super User to set access to a given class of documents within the domain allowed. A typical user can have two different kinds of access. Any document should be available for any user for observation and study but only certain documents are available for modification by a given user. In this way a user, can copy and insert information from a higher priority document into the document he/she is working modifying but will be unable to modify the higher level document.

CAPS User/System Interface

The following sections describe the design for the CAPS interface. The figures included here illustrate the design as it would be implemented on the limited-size screen of an Apple Macintosh. However, the workstations to be used in CAPS will be larger, thus allowing larger document windows, more side-by-side documents, etc.

Figure 31 illustrates the basic CAPS user screen. The menu line at the top of the screen indicates the various options available to the user. These include: Utilities (under the CAP symbol), USER, FILE, EDIT, HELP, ACCESS, COMPONENTS, and PRIORITY. The CAPS system user would greet an empty screen with only the USER option highlighted (Figure 32).

Selecting USER option provides the pull-down menu shown in Figure 33. Only the Identification option would be available until the user has entered his ID. This is a priority system, hence until the user enters his/her ID none of the options of the system are available. Selecting Identification from the pull-down menu displays the dialog box shown in Figure 34. The user types in his/her ID in the proper place. If the system does not recognize the ID, then the message shown in Figure 35 is displayed.

The user is unable to use the system until someone with priority clearance uses the system to register the user. A priority user, known as a Super User, accesses the system in the same way as any other user except that the system recognizes the ID number as a person with priority clearance. As soon as the ID is recognized for a Super User, the PRIORITY option on the menu bar is included. For non-priority users, this item does not appear on the menu.

Selecting the PRIORITY option provides the pull-down menu shown in Figure 36. The Super User selects Access Control to register a new user, delete an old user, change the priority or access of an existing user, or to change his/her own ID number to provide the necessary security. Selecting Access Control displays the menu shown in Figure 37. Selecting Register a New User displays the dialogue box shown in Figure 38. The Super User can type in the new user name, an appropriate ID number or word, set the priority level of the user, and indicate the documents to which the user may have access. A similar dialogue box is provided when the priority user is changing the access or priority of an existing user.

The Template Editor is accessed from the PRIORITY pull-down menu by an authorized User. The Template Editor allows the system to be continually modified to reflect the latest in doctrine and procedure, not only in the training documents produced by the system but also in the nature of the documents and their standarized format. The details of this Template Editor will have to wait for the formal design of the system but include the following:

Each training document will have a specialized input format. The information requested can be modified by the Template Editor. This information is modified by the Translator to provide the report form of the training documents with a minimum of effort by the user. The Template Editor will also

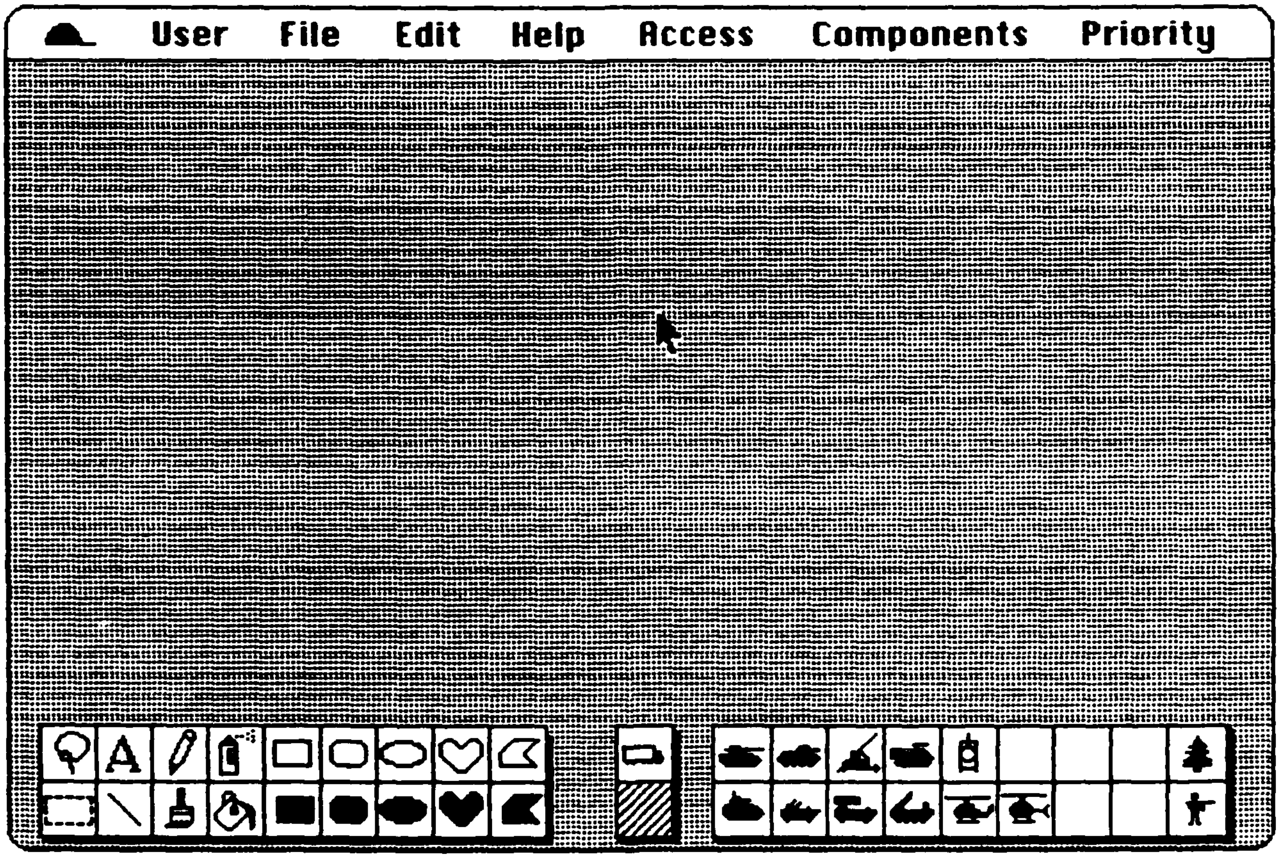


FIGURE 31. CAPS User Screen

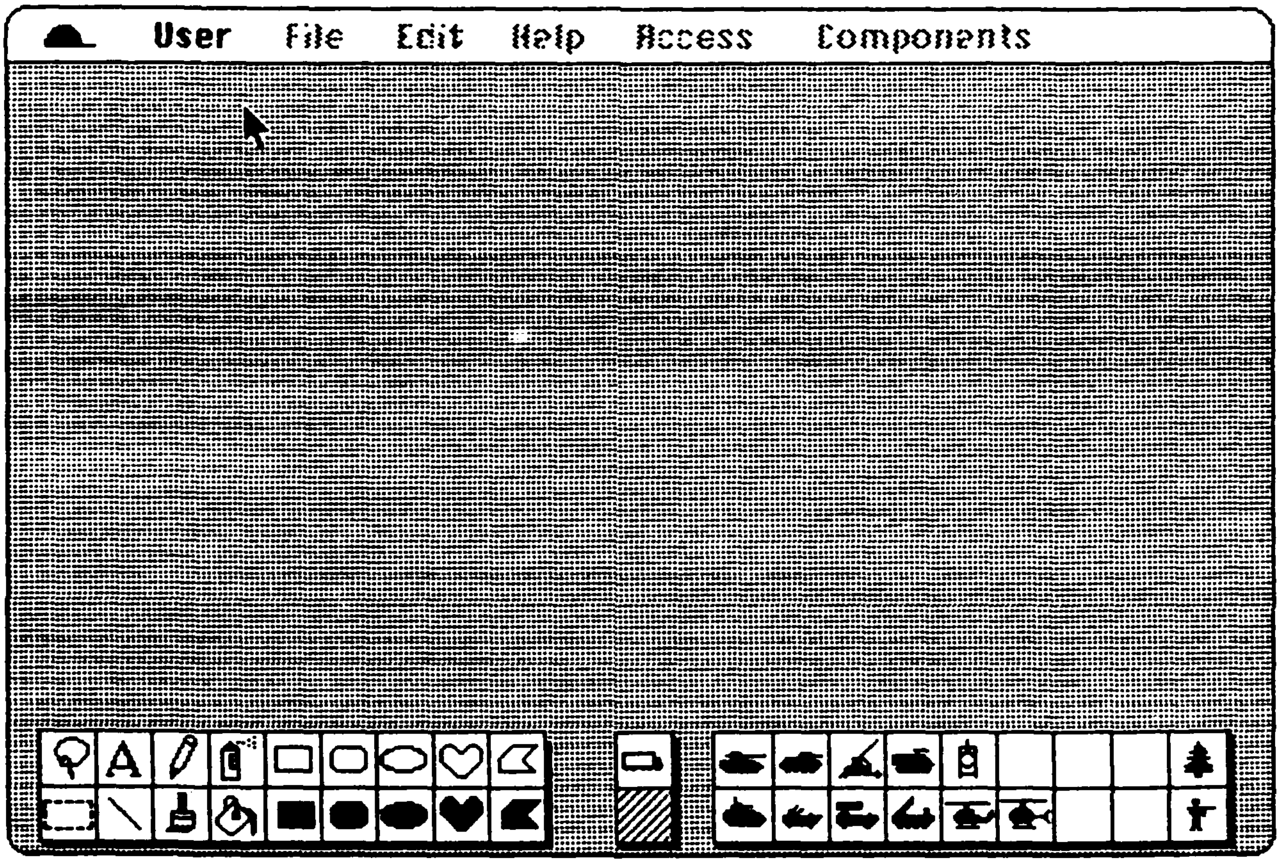


FIGURE 32. CAPS User Screen - Initial State

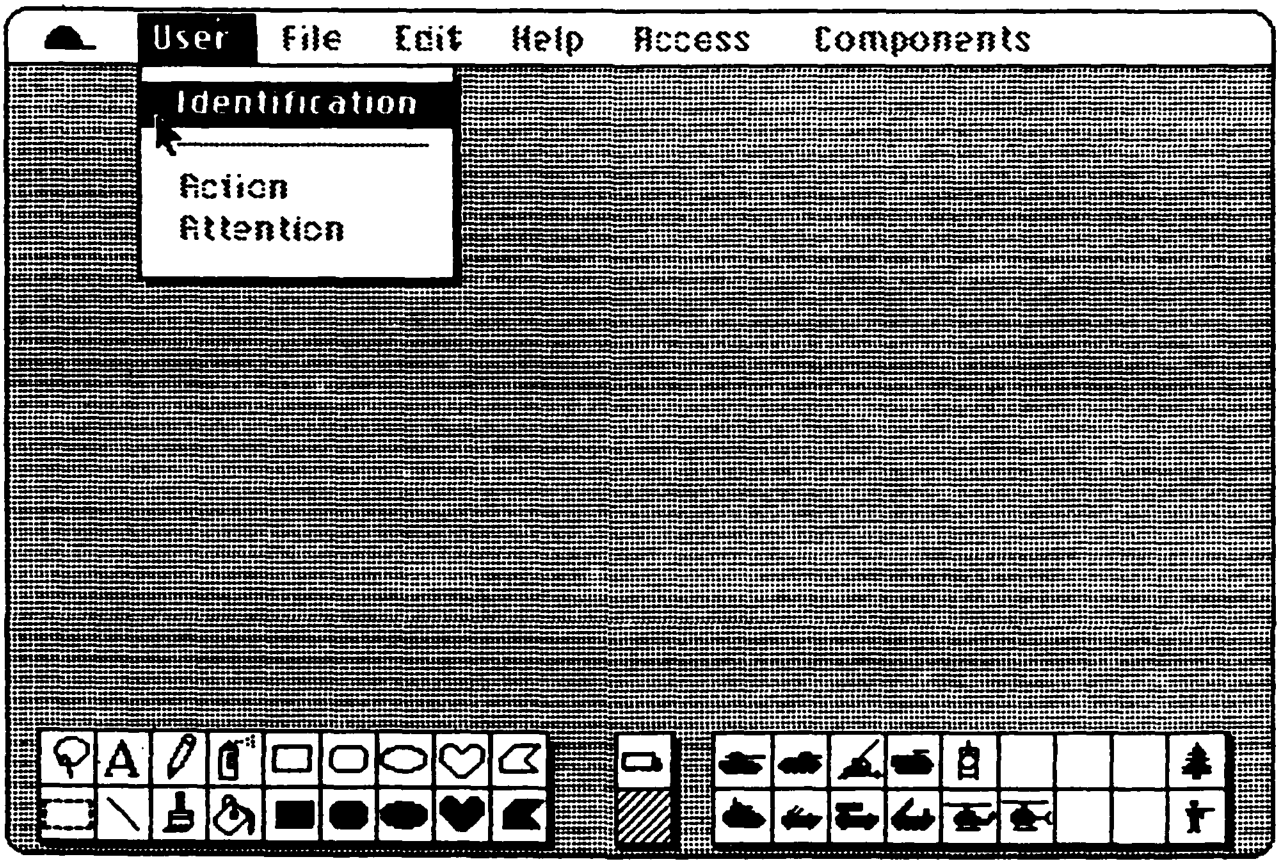


FIGURE 33. User Pull-Down Menu

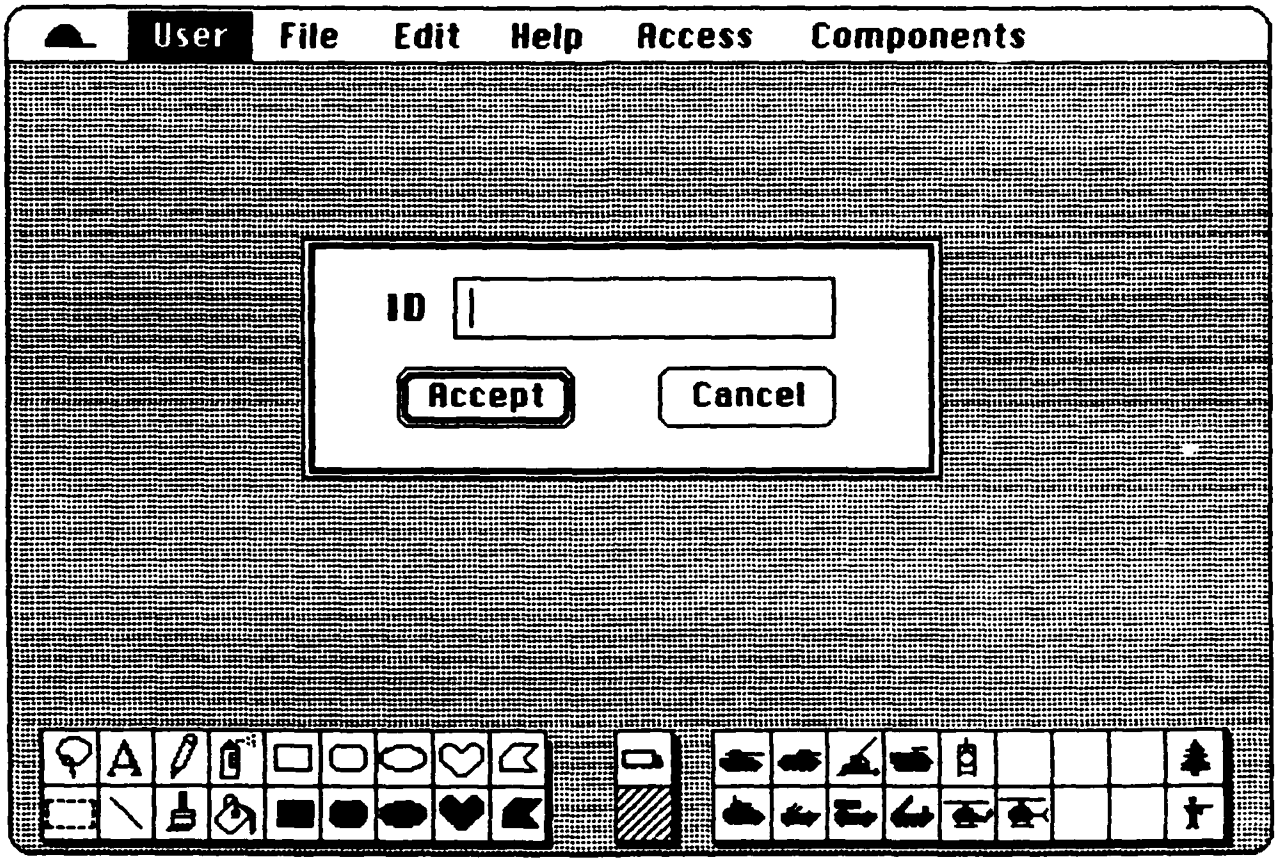
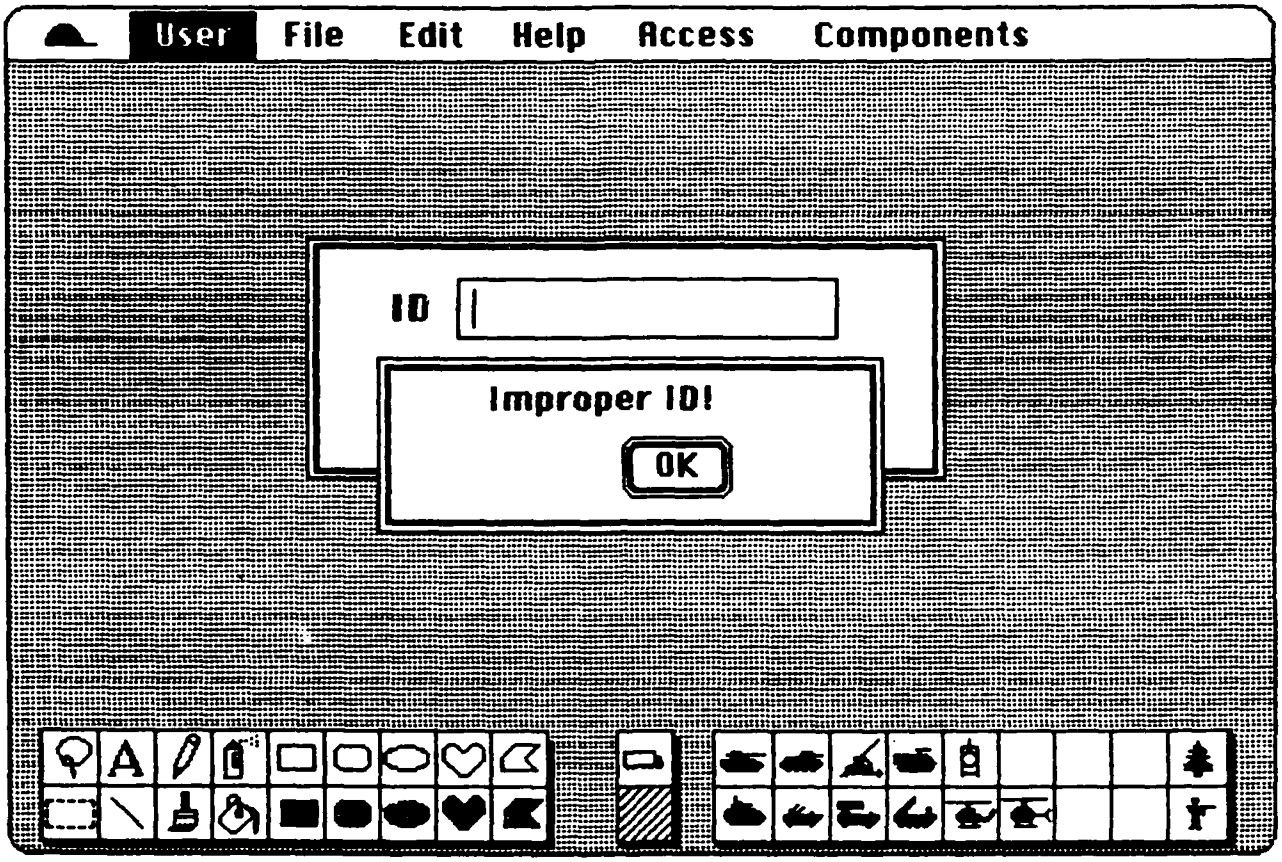


FIGURE 34. USER Identification Dialog



Improper ID!

FIGURE 35. Improper ID. Dialog

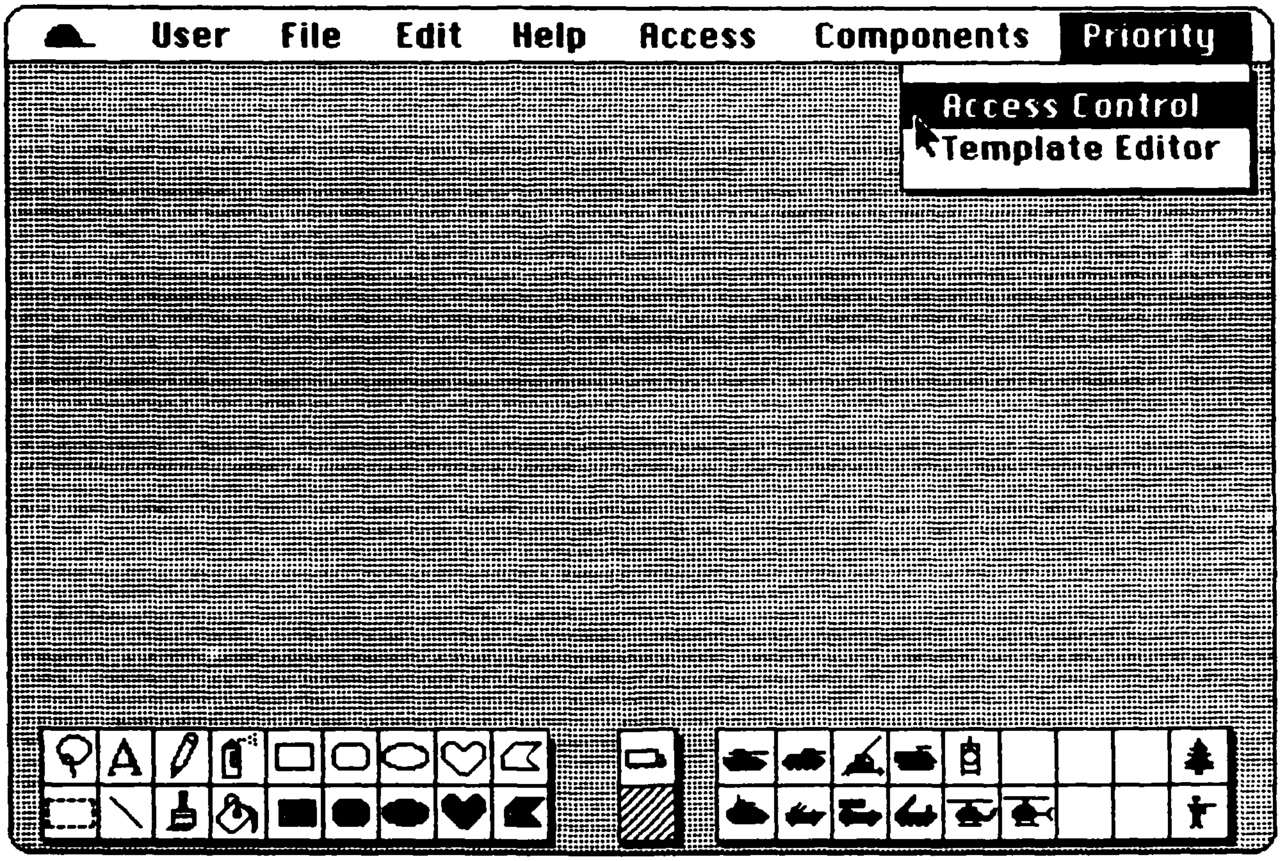


FIGURE 36. Priority Pull-Down Menu

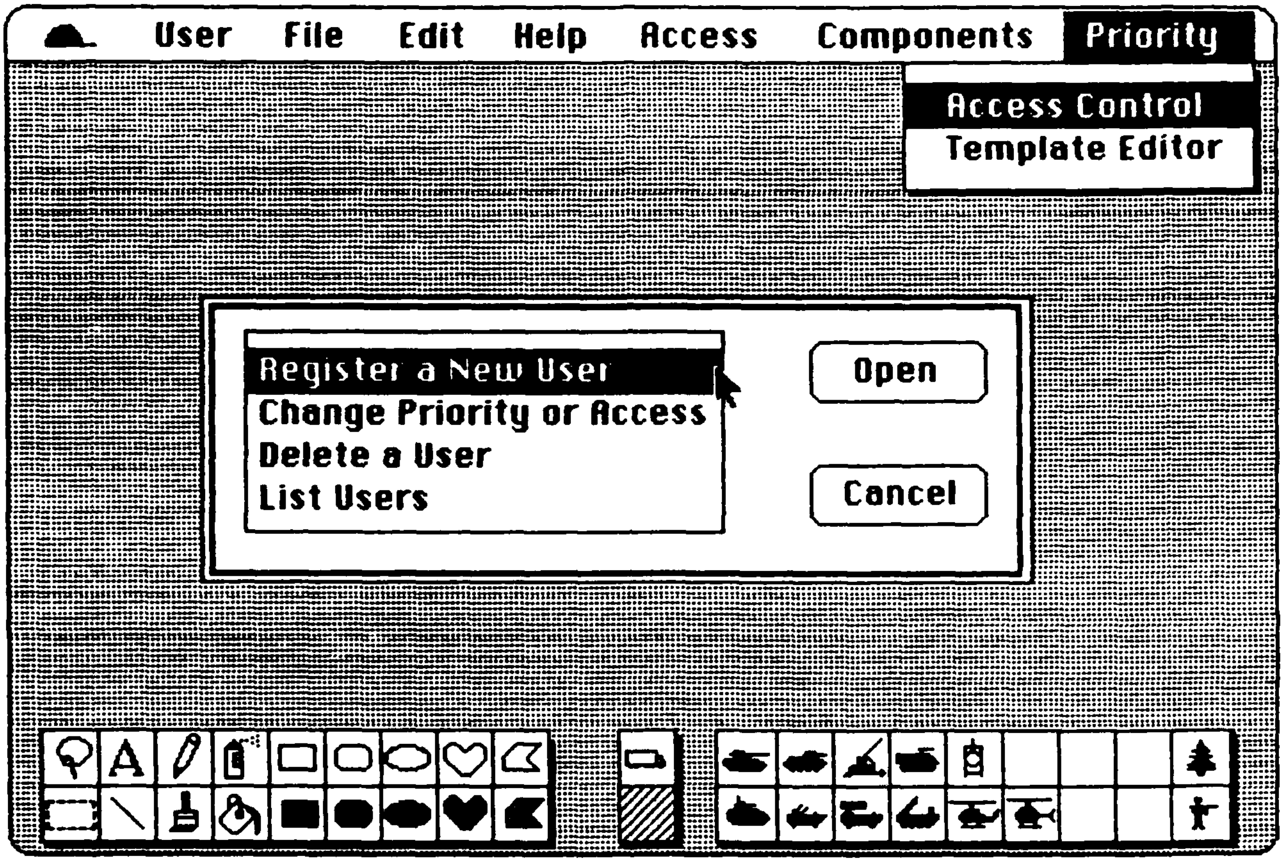
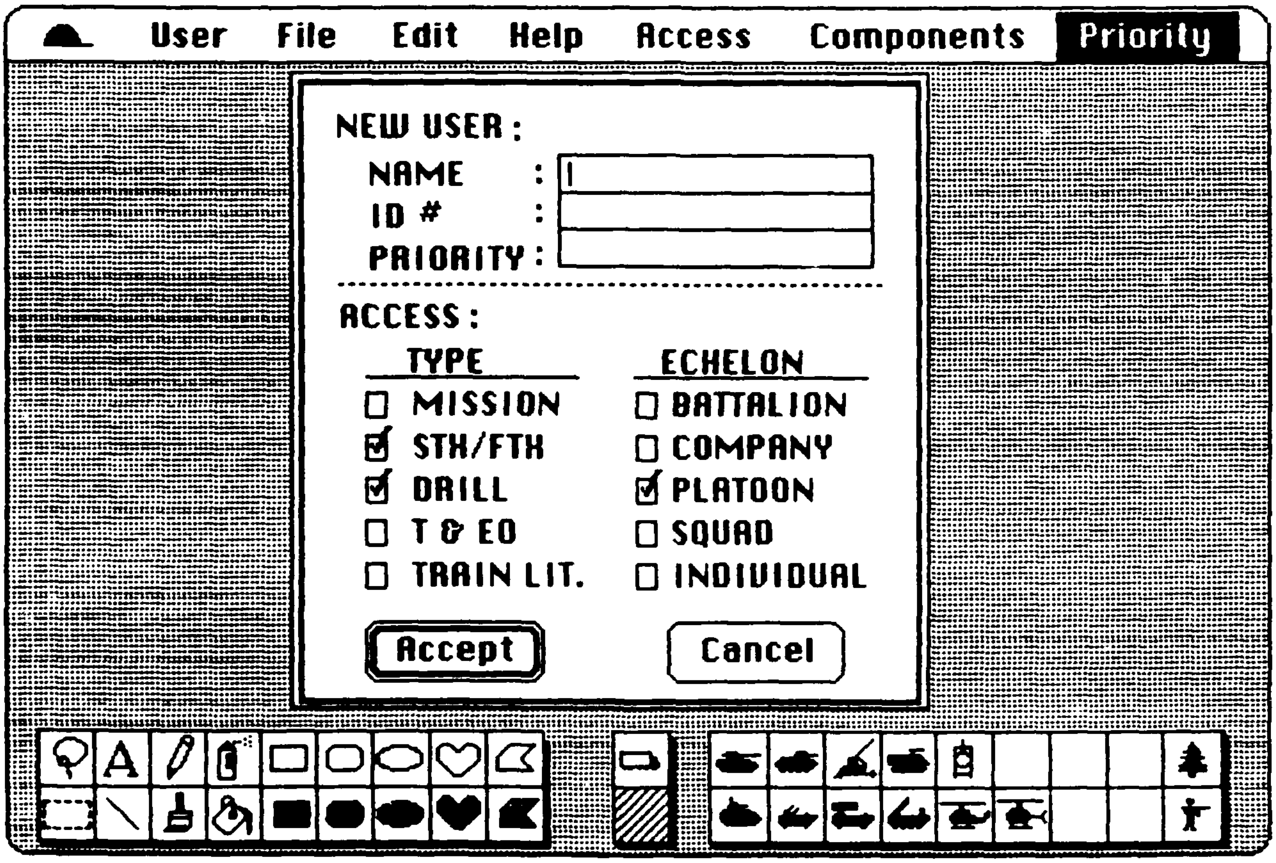


FIGURE 37. Access Control Dialog Menu



NEW USER :

NAME :

ID # :

PRIORITY :

ACCESS :

TYPE	ECHELON
□ MISSION	□ BATTALION
☑ STH/FTH	□ COMPANY
☑ DRILL	☑ PLATOON
□ T & EO	□ SQUAD
□ TRAIN LIT.	□ INDIVIDUAL

FIGURE 38. New User Dialog

enable a Super User to modify the way that the Translator modifies this information. The system also includes an "intelligent" Checker, which checks the input information and the information placed in reports. The Template Editor enables the priority user to modify the nature of this "intelligence" by changing the expert model included to reflect the latest changes in doctrine. The Template Editor will also enable the priority user to change the Guidance and Instruction provided from the HELP option on the menu bar. The Template Editor will enable the system to remain current and to be modified to reflect changes in the way training is conducted without making it necessary to make changes in the software and hardware comprising the CAPS system.

Once the user has been identified to the system, the other menu items become available for the user. To begin work on a document the user must first set the proper access using the ACCESS option from the menu bar. Selecting ACCESS shows the pull-down menu shown in Figure 39. Only those items to which the user has authorized access will be available. The top set of items indicate the type of document to be used; the bottom set of items indicate the Echelon level of this document. The check marks indicate the current selection.

Having selected the appropriate access, the next step for the user is to open the document for work. The user selects Open from the FILE pull-down menu, as shown in Figure 40. A number of other convenient facilities are available on the FILE pull-down menu. The Save command enables the user to save the result of modifications in the document that is currently being created or modified. CAPS will have an automatic back up system that will always keep the new document or document modification, as well as the past two versions of the document. The Save As... command enables the user to save a temporary version of the document while work is in progress. This command does not allow the user to create alternate versions of a given document as part of the permanent database, but does enable the user to create working versions of a given document while modifications are in progress rather than leaving the database in a partially modified state. In other words, the user can call up a document from the database, use the Save As... command to create a temporary file while modifications are in progress. Then when the modifications are complete and approved by the appropriate authority, the temporary file can be used to replace the current version of the permanent file.

The Print command allows the user to print the current version of the document. The New command allows the user to create temporary files as necessary but will not allow such a temporary file to be saved as a permanent file.

Selecting Open command from the FILE pull-down menu provides the dialog box shown in Figure 41, which lists all of the documents of the type and echelon selected. With this dialog, the user can select an Input format for the document or a Report format. The Input format is like a database requesting specific information from the user so that the computer can construct the report from the information provided. There will be unique input formats for each of the document types in the system. These will be forms on the screen that require the user to supply the necessary information. Redundant information that can be generated by the system will be provided

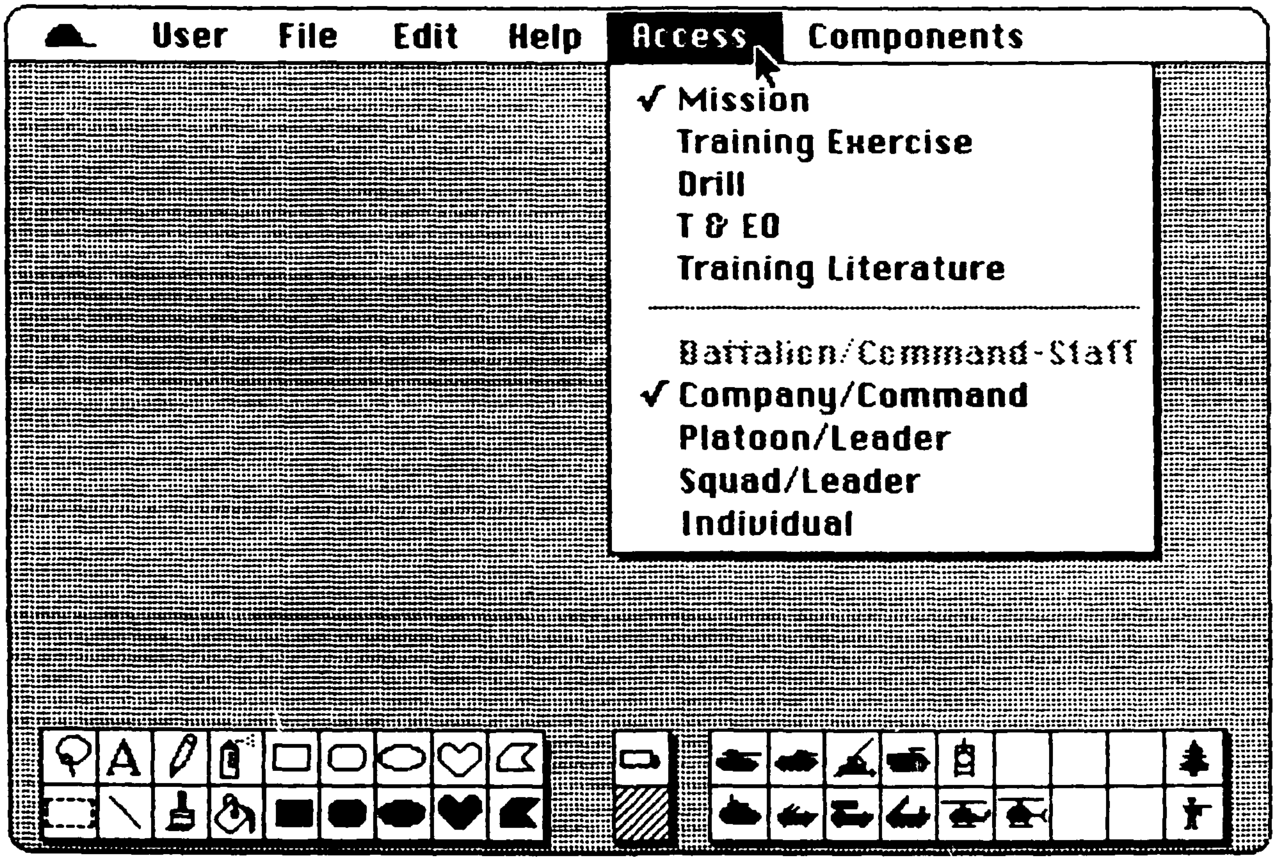


FIGURE 39. Access Pull-Down Menu

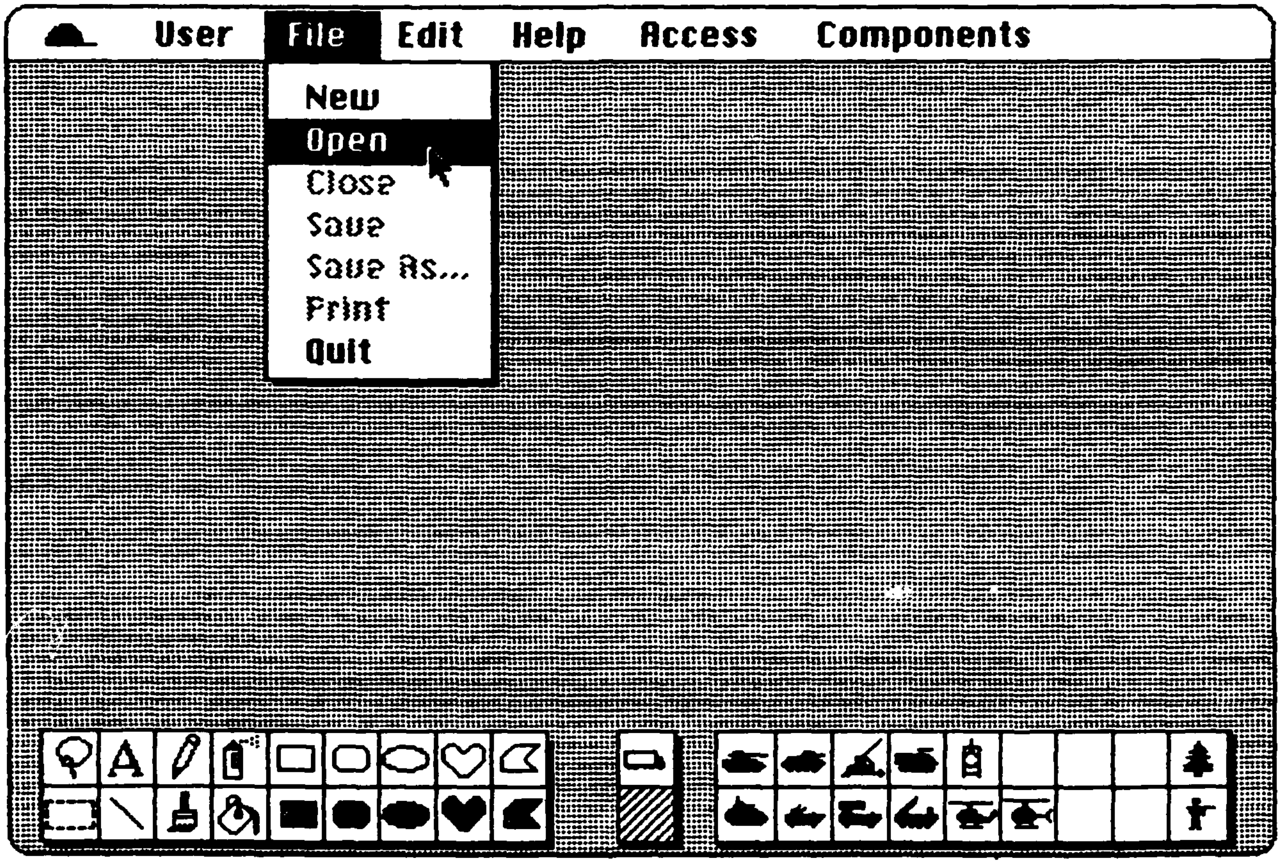
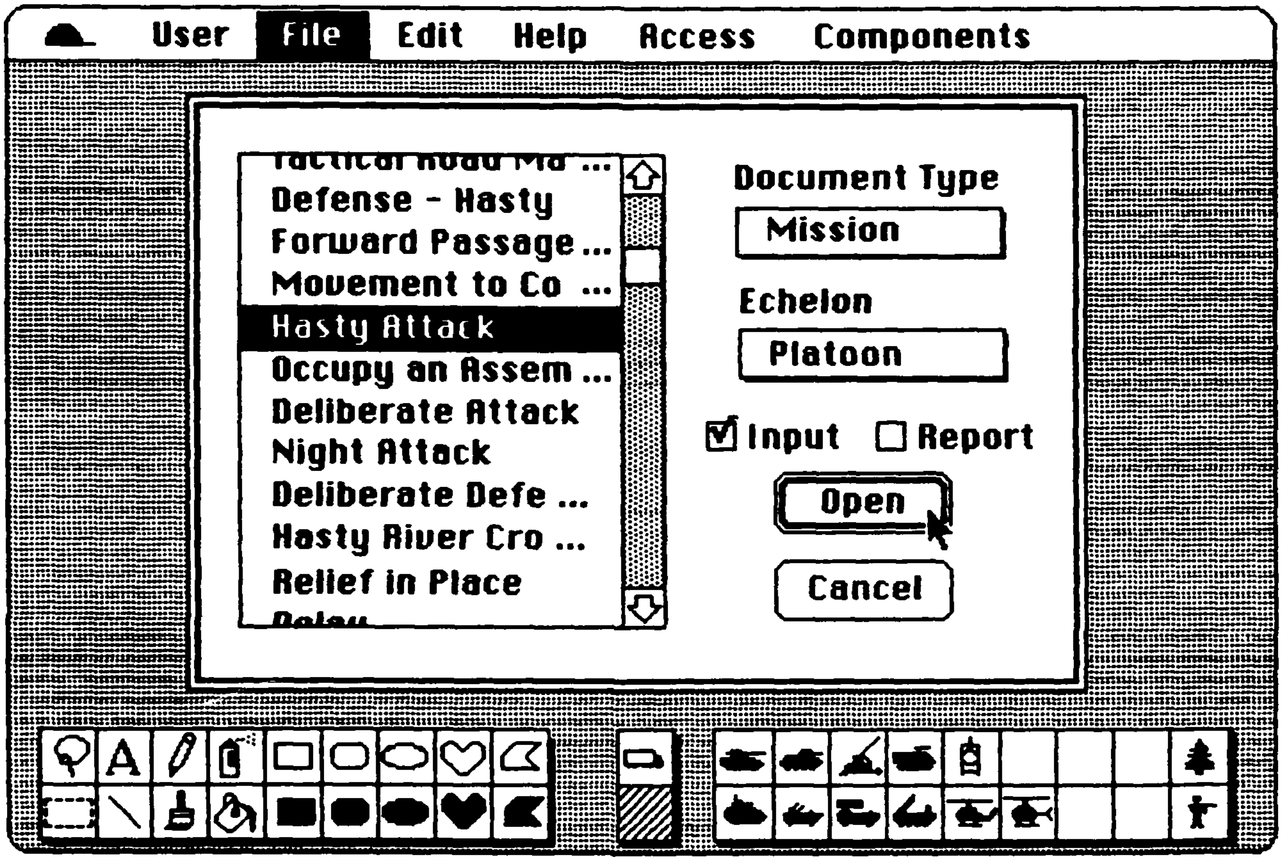


FIGURE 40. File Pull-Down Menu



Tactical Road Ma ...
Defense - Hasty
Forward Passage ...
Movement to Co ...
Hasty Attack
Occupy an Assem ...
Deliberate Attack
Night Attack
Deliberate Defe ...
Hasty River Cro ...
Relief in Place
Delay

Document Type: Mission

Echelon: Platoon

☑ Input ☐ Report

FIGURE 41. Document Select Dialog Menu

automatically. The exact format for these input forms will be designed as part of the system design at a later time. The user selects a document for work by *clicking the name and then clicking the Open box. The selected* document appears on the screen in an open Document Window, as shown in Figure 42. Note that the illustrated document contains both graphic and text information.

The Report format shows the document as it will appear in the training literature being prepared. The Input format assists the user by providing guidance as to what is required but after the user has input the necessary information the Translator function converts this input into a Report format. In Report format, the user can edit the document like any other document.

Edit commands are available under the EDIT pull-down menu, shown in Figure 43. (Note that the pull-down menu is superimposed over any existing information on the screen, in this case the open document.) The Copy, Cut, and Paste enable the user to move blocks of text or graphics from one document to another or from one section of a document to another.

A clipboard holds information from the last cut or copy until it is pasted. The system will also have a note pad where the user can keep memos to him/herself. The system will also have a scrapbook where a series of figures or pieces of text can be kept for convenient access when creating or modifying a document.

The Undo enables the user to abort the last operation. Show Clipboard enables the user to examine the last item cut or copied. As with all other CAPS menu functions, these functions will be available from the keyboard, as well as from the Edit menu.

The EDIT menu also includes some special functions that enable it to act like a combined word processor (writing) and graphics processor (drawing). The Font option enables the user to select from the dialog box shown in Figure 44. The user can select the type, size, and style of font. A variety of fonts, sizes, and styles can be used on any document as required.

The Graphics option is a toggle that displays the graphics menu and icon options at the bottom of the user's screen. The left set of options include the lasso for moving items without the background, the selection rectangle for moving items with the background, the letter for selecting text from graphics mode, *the line segment, a pencil, a paint brush (with various sizes), the* spray can, and the paint bucket to fill areas. The shape drawing functions are also included in the left set of options. The items on the right have two modes: pattern mode and icon mode. In pattern mode (selected by clicking the pattern box in the middle) a variety of fill patterns is displayed, as shown in Figure 45. In Icon mode, the boxes contain commonly used pictures necessary for diagramming battle plans and other training information. These can be selected by dragging the icon to the document. They can be enlarged or made smaller by the selection rectangle. There may be several icon menus selected by double clicking the icon selection box.

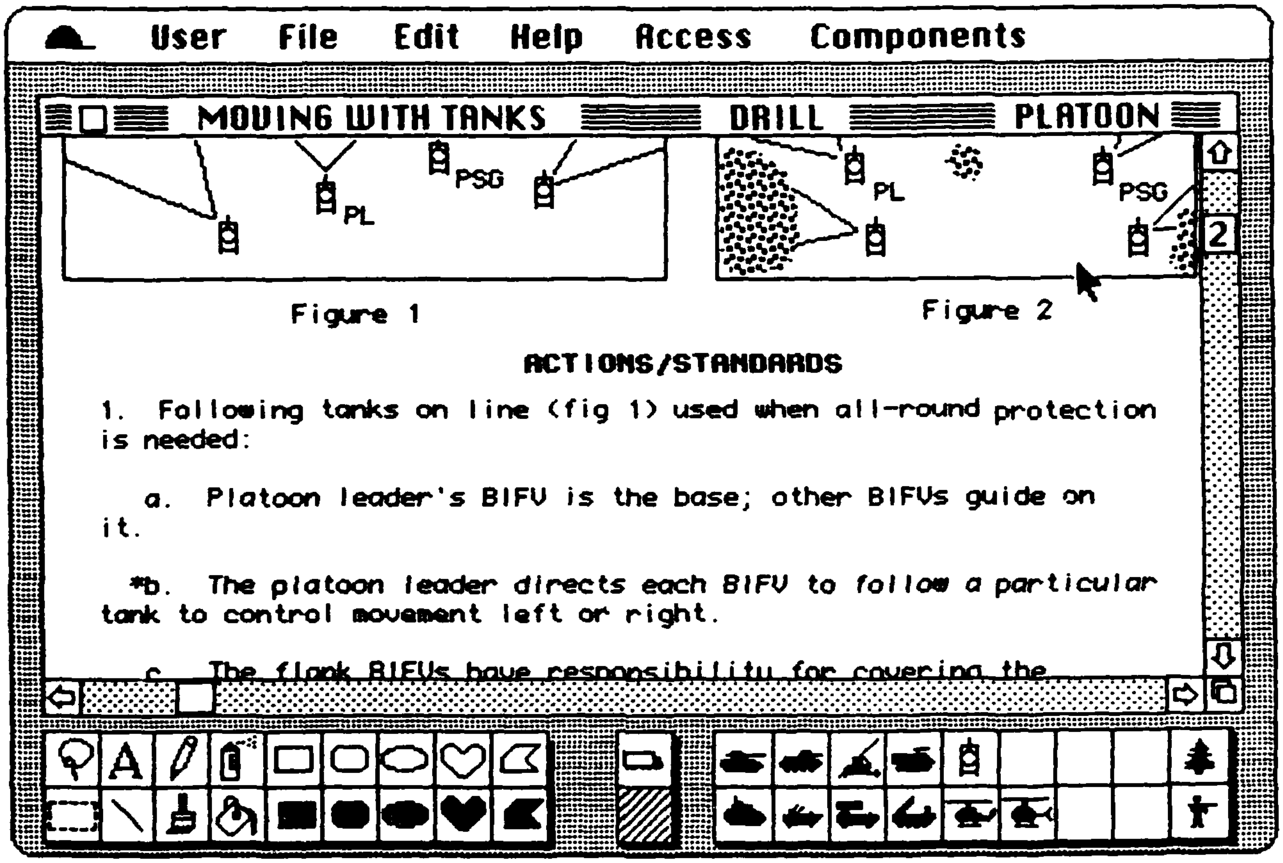


Figure 1

Figure 2

ACTIONS/STANDARDS

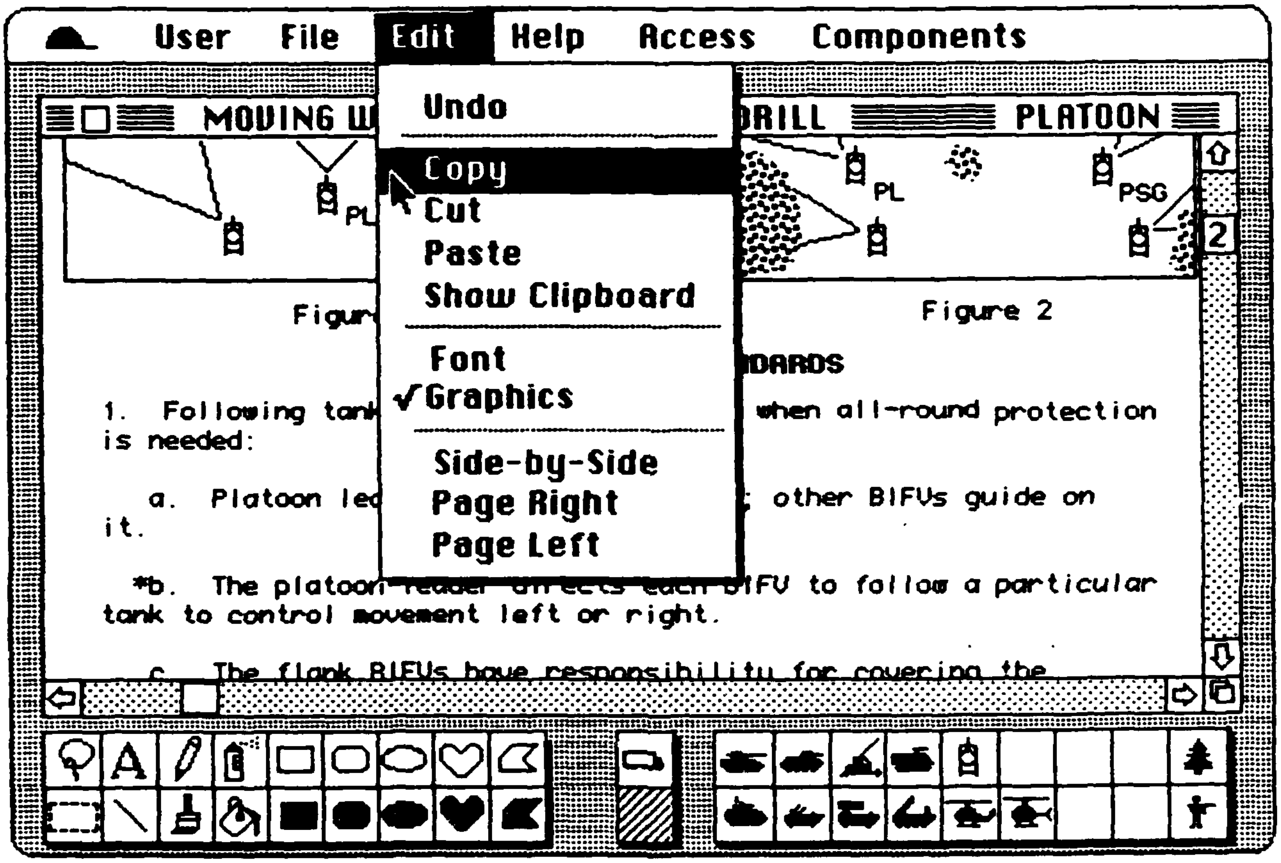
1. Following tanks on line (fig 1) used when all-round protection is needed:

a. Platoon leader's BIFV is the base; other BIFVs guide on it.

*b. *The platoon leader directs each BIFV to follow a particular tank to control movement left or right.*

c. The flank BIFVs have responsibility for covering the

FIGURE 42. Open Document Window



Figur

Figure 2

DARDS

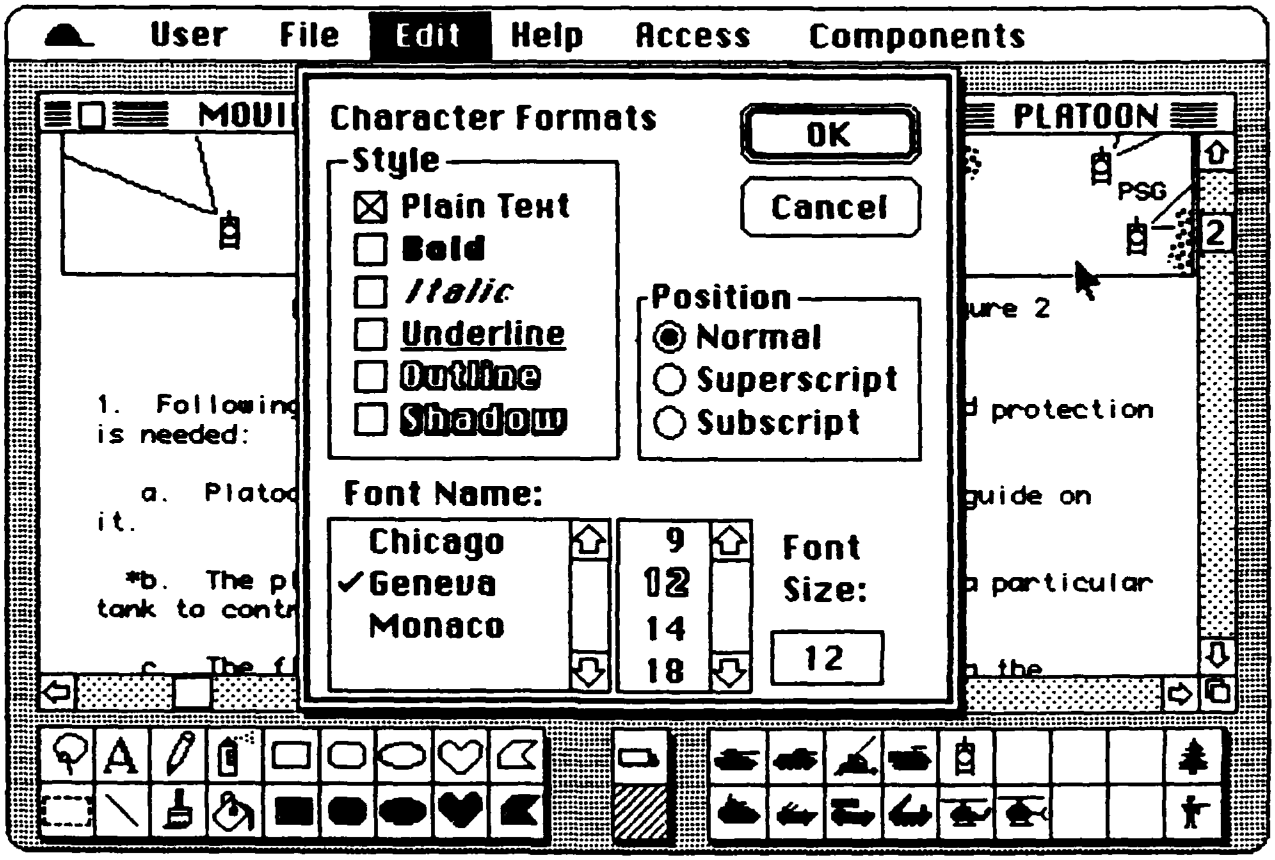
1. Following tank when all-round protection is needed:

a. Platoon le other BIFVs guide on it.

*b. The platoon leader directs each BIFV to follow a particular tank to control movement left or right.

c. The flank BIFVs have responsibility for covering the

FIGURE 43. Edit Pull-Down Menu



Character Formats

Style

☒ Plain Text
□ Bold
□ Italic
□ Underline
□ Outline
□ Shadow

Position

● Normal
○ Superscript
○ Subscript

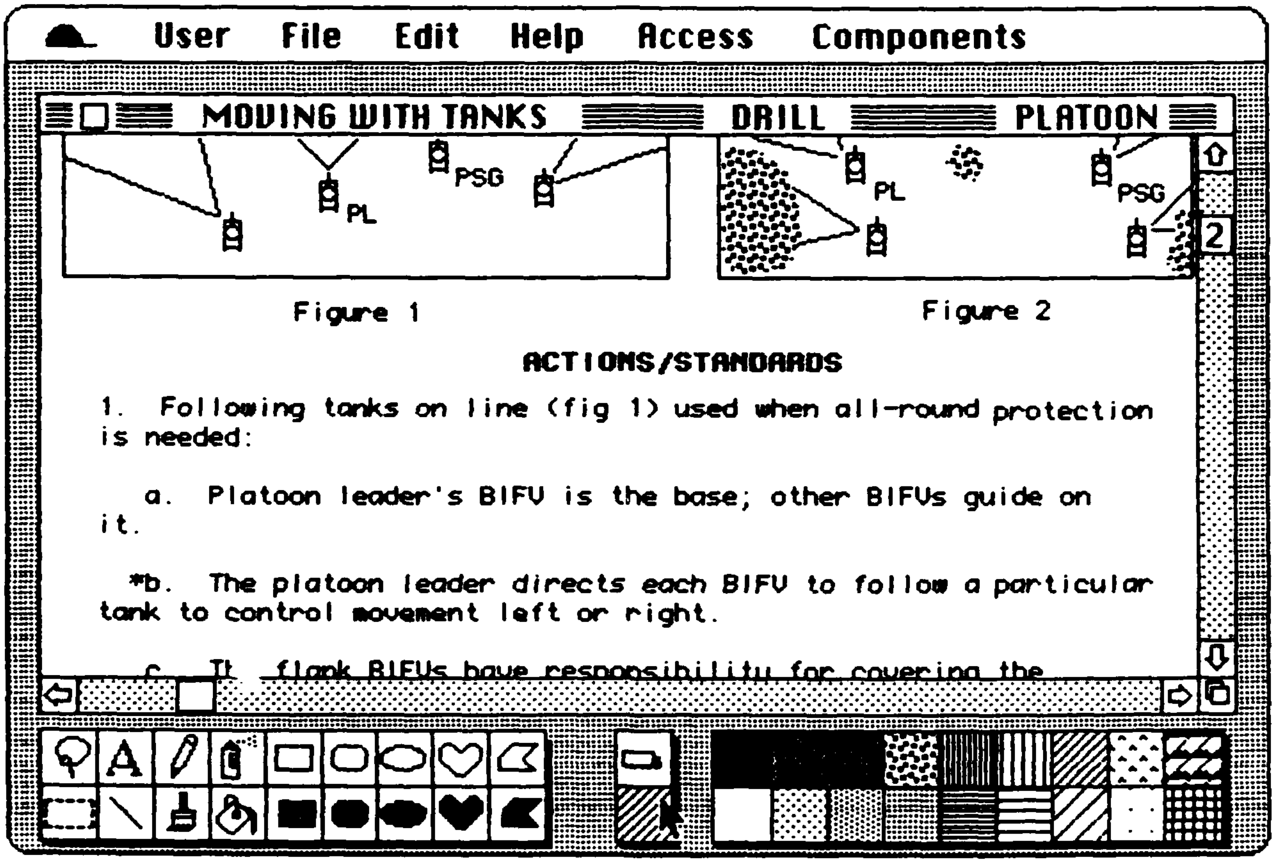
Font Name:

Chicago
✓Geneva
Monaco

9
12
14
18

Font Size: 12

FIGURE 44. Font Dialog



ACTIONS/STANDARDS

1. Following tanks on line (fig 1) used when all-round protection is needed:

a. Platoon leader's BIFV is the base; other BIFVs guide on it.

*b. The platoon leader directs each BIFV to follow a particular tank to control movement left or right.

c. Th flank BIFVs have responsibility for covering the

FIGURE 45. Pattern Mode

The CAPS system will allow the user to open as many documents as may be necessary to enable the user to work on more than one document at a time and to cut and paste between documents without going through the scrap book. (There will probably be some limit on the number of open documents but it will be sufficient to enable the user to use the system efficiently.) The EDIT menu includes a special feature which allows the user to either overlay document on the same screen or to place open documents side-by-side. In the Side-by-Side mode, opening a second document places it to the right of the current document by scrolling the entire screen. The Page Right and Page Left commands enable the user to scroll the various open documents and to have each occupy the entire screen. In either overlay or side-by-side mode, the user can change the size of a given document as necessary.

The utility options are available from the CAPS pull-down menu. As shown in Figure 46. These include the Scrapbook, the Note Pad, Calculator, Calendar, and the ability to adjust the system for personnel use via the Control Panel.

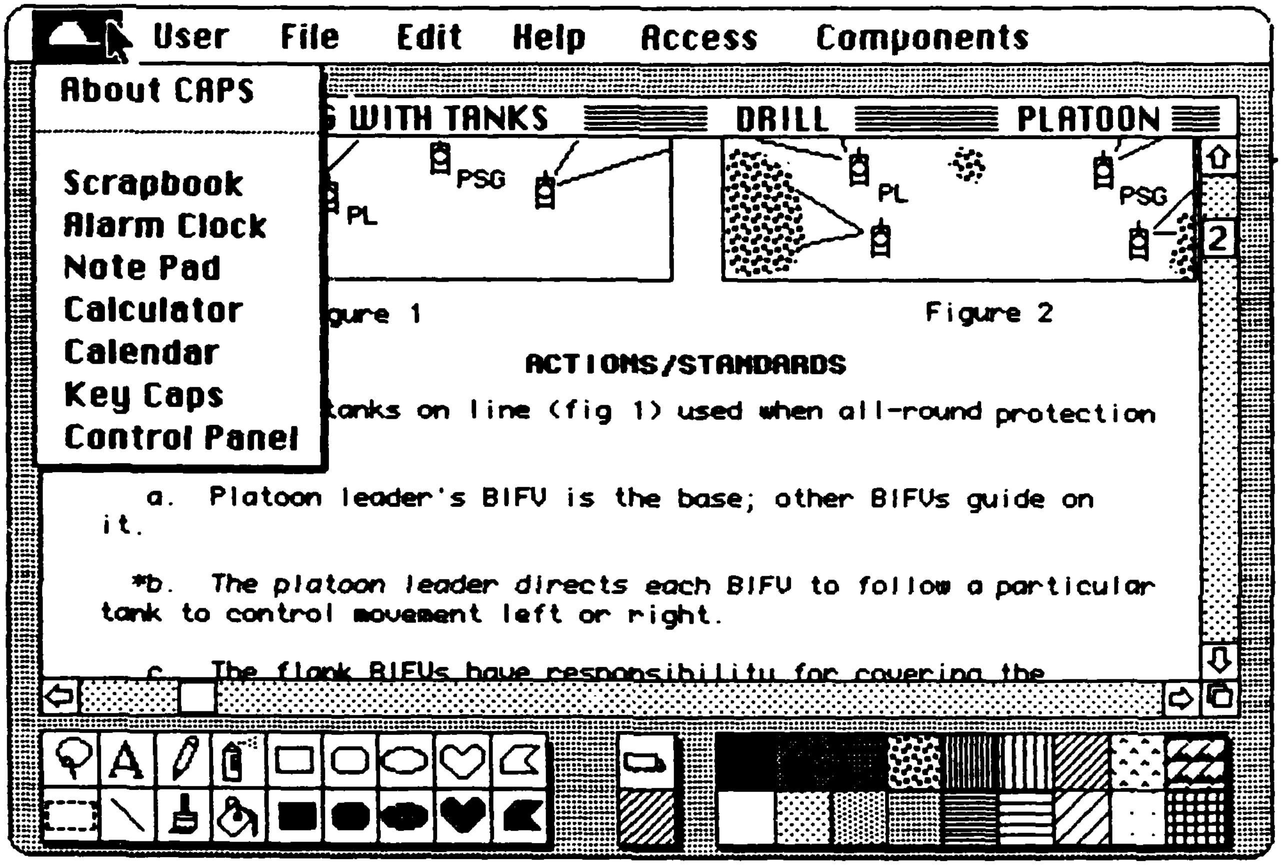
The CAPS system will be designed for the experienced user and thus efficient to use. For the less experienced user, the HELP pull-down menu is available. As shown in Figure 47, four levels of on-line help are proposed. Guidance provides a set of menu choices for the user in the form of question decision trees to enable the user to be led to provide the necessary information into the input forms. This option is a toggle which is either switched on or off as the user may desire. When Guidance is ON, the user is provided considerable information to assist him/her by supplying the necessary information. The exact form of this assistance will be developed as part of the formal design of the CAPS system.

The Instruction option is a form of hypertext. The user can request Instruction for any item or for an entire form or document. The system then opens a Instruction window that tutors the user in the background and the information requested. This help accompanies the document being developed in a side-by-side or overlay window so that the user can move easily between the help and the application.

The Doctrine/Regulations option is also a form of hypertext, which shows the user all of the doctrine and regulations pertinent to the item(s) shown in the active Document window.

The Roadmap option is a graphic representation of the underlying CAPS database, which shows the user which document(s) are currently being modified and where these documents appear in the overall ARTEP database. The actual content of the HELP options will be developed as part of the formal system design.

The COMPONENT pull-down menu, shown in Figure 48, is the way that the user gains access to the cross reference matrices, described earlier (CAPS Data Representation). By selecting this item, a user can determine the location and composition of any training document that may be under consideration. New training documents can only be added to the system by first locating the appropriate location for the new document in the component matrix and adding a name and identification number for this new component. The component will



ACTIONS/STANDARDS

tanks on line (fig 1) used when all-round protection

a. Platoon leader's BIFV is the base; other BIFVs guide on it.

**b.* *The platoon leader directs each BIFV to follow a particular tank to control movement left or right.*

c. The flank BIFVs have responsibility for covering the

FIGURE 46. Utilities Pull-Down Menu

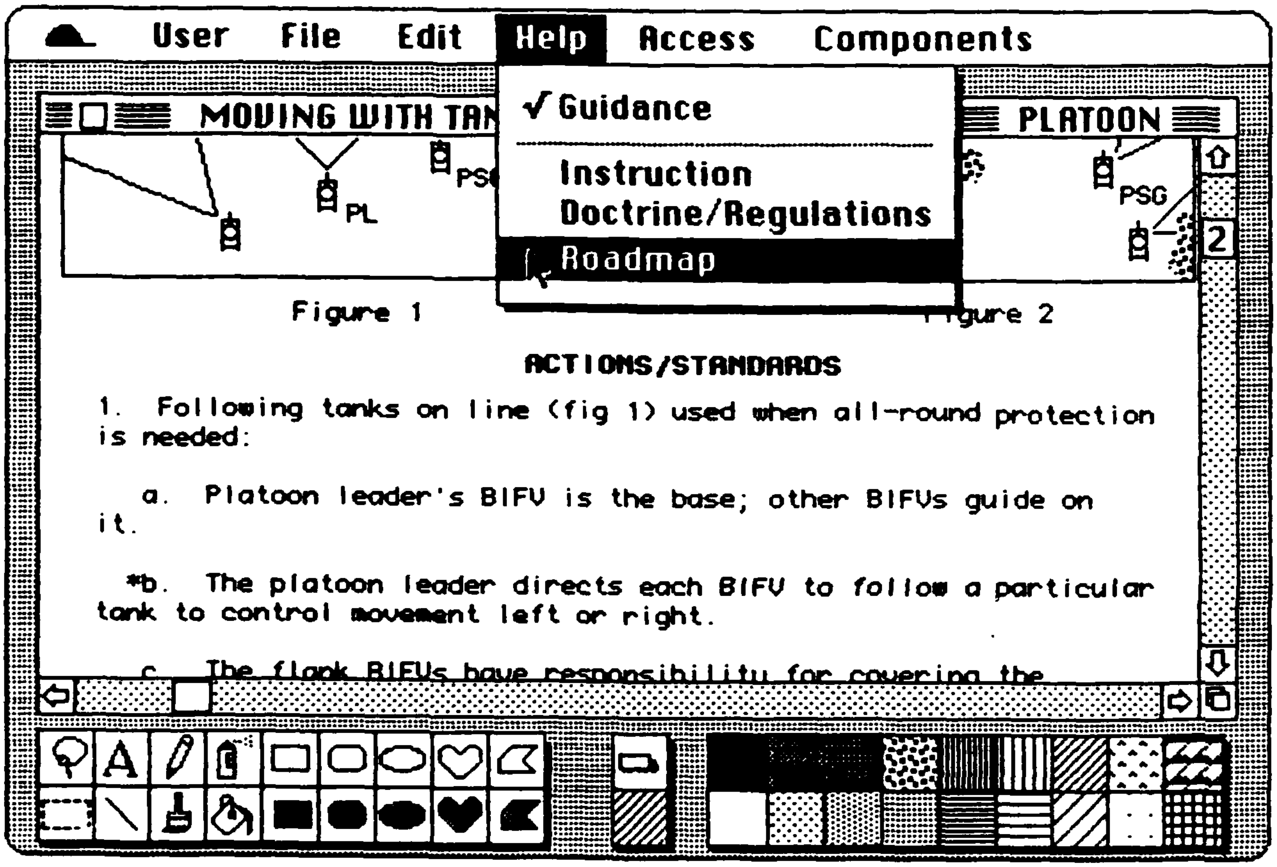


Figure 1

Figure 2

ACTIONS/STANDARDS

1. Following tanks on line (fig 1) used when all-round protection is needed:

a. Platoon leader's BIFV is the base; other BIFVs guide on it.

*b. The platoon leader directs each BIFV to follow a particular tank to control movement left or right.

c. The flank BIFVs have responsibility for covering the

FIGURE 47. Help Pull-Down Menu

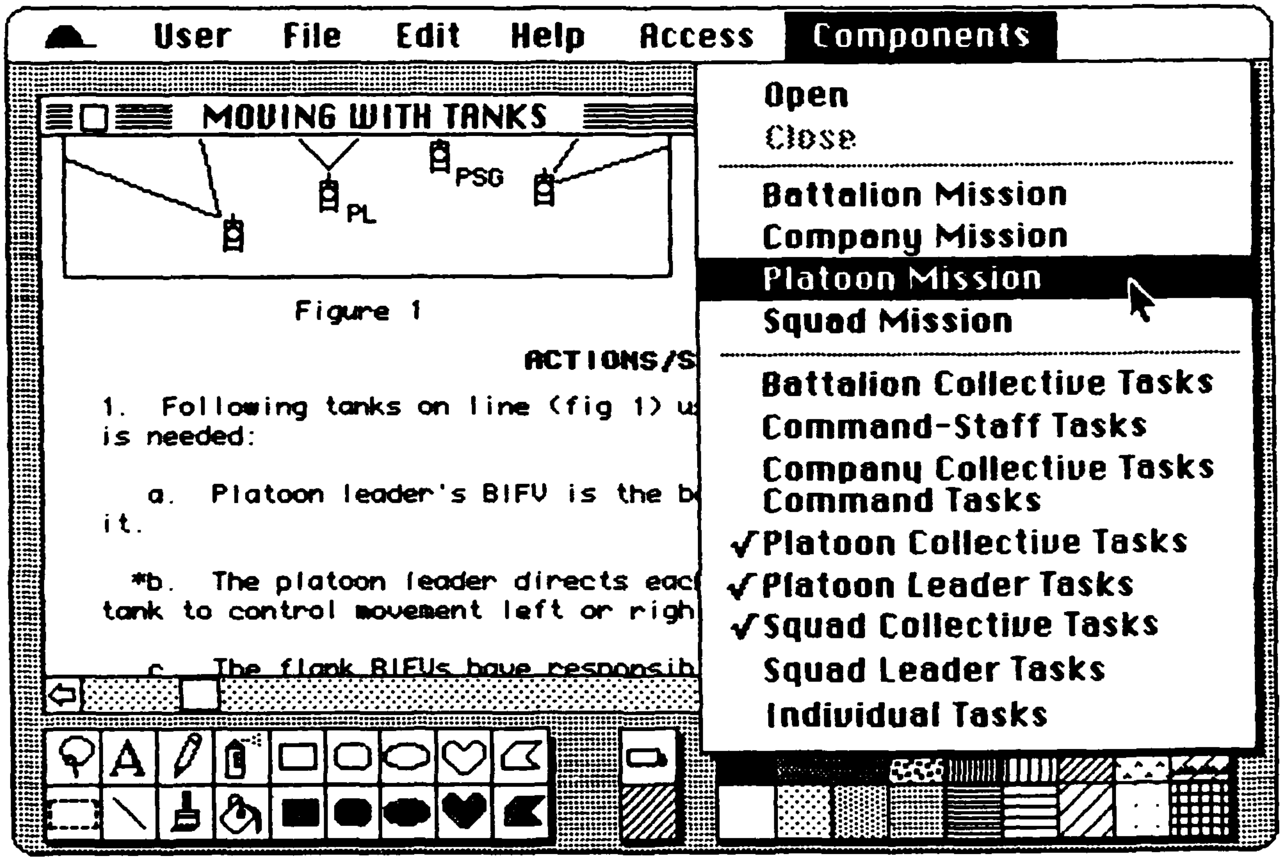


Figure 1

ACTIONS/S

1. Following tanks on line (fig 1) u
is needed:

a. Platoon leader's BIFV is the b
it.

*b. The platoon leader directs eac
tank to control movement left or righ

c. The flank BIFVs have responsib

FIGURE 48. Component Pull-Down Menu

then be available to Open from the FILE pull-down menu. Old training documents can only be deleted from the system by first locating the document in the cross reference matrices and eliminating its title.

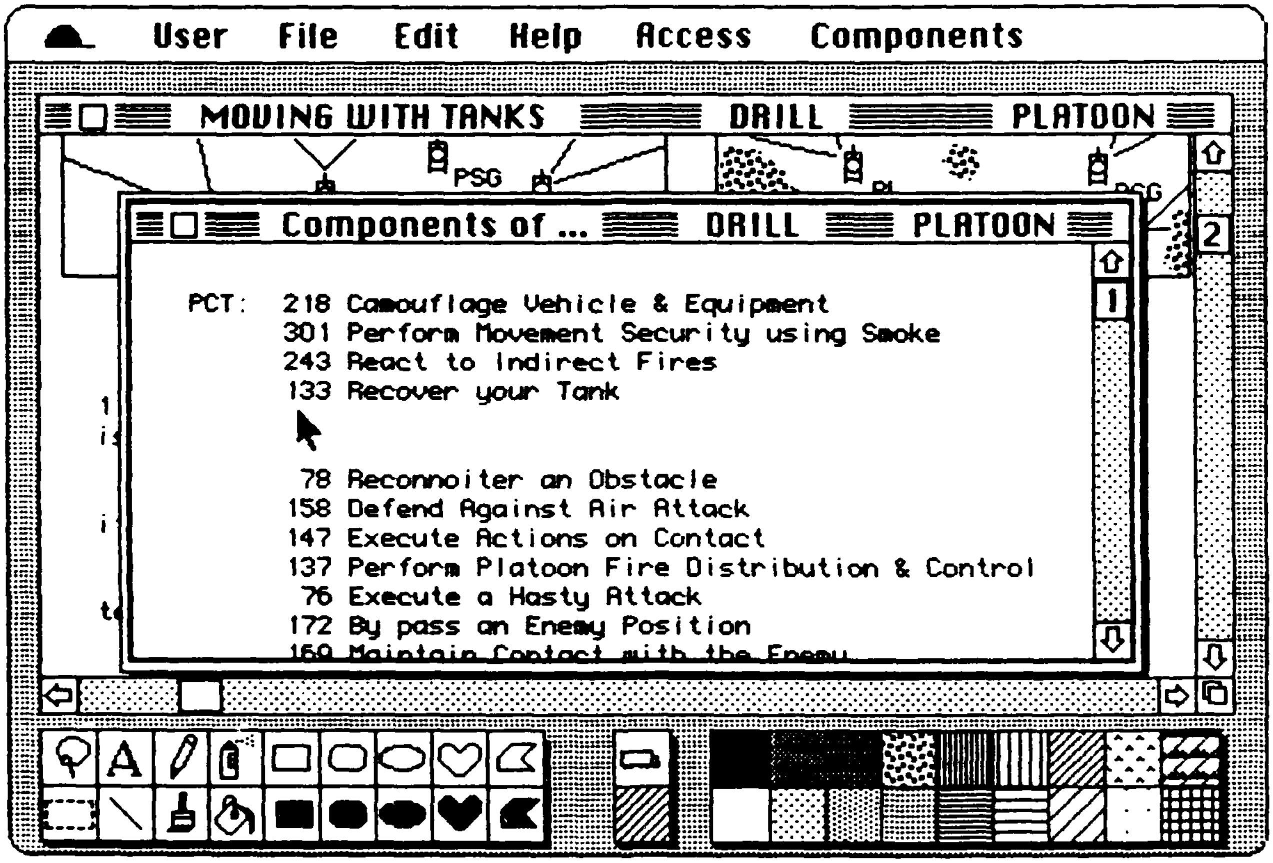
The components of the system are all components of training missions. To use this menu, the user must first select the echelon of the mission desired. The system defaults to the mission corresponding to the current document. A different mission level can be selected by clicking the appropriate mission level on the COMPONENTS menu. The user then selects the amount of detail desired from the lower part of the COMPONENTS menu. The available choices are determined by selection of the mission. The available mission level is determined by the user access and priority. Only those options below the selected mission are available to the user. The user can select one or more levels of detail, provided that to select a lower level the user must also select the higher level. For the menu shown previously in Figure 4-19, the mission is "Platoon." Available options include tasks from the Platoon Collective Tasks level down to Individual Tasks. The User can select any number of options from Platoon Collective Tasks down. However, the user cannot select Individual Tasks without first selecting Squad Collective Tasks. Similarly, he cannot select Squad Collective Tasks without first selecting Platoon Collective Tasks.

The user opens the cross reference document by selecting Open from the COMPONENT menu. Selecting this option provides a dialog box as shown in Figure 49. The names in the scroll box are all of the missions for the echelon shown. This component dialog menu is used to add or delete tasks from the system. The user adds a task by inserting the task name and identification in the appropriate place in the outline. The user deletes a task name by dragging the task name and selecting Cut from the EDIT pull-down menu. The edit commands are available like any other document.

A dialog box (not shown) is displayed whenever a user attempts to add or delete a document. This box asks the user to provide additional authorization information, indicating that there has been an action request and that this user is the proper person to make the addition or deletion. The user can only add or delete tasks in those areas where his/her priority allows, but any user can view the cross reference information for any part of the system. Whenever a task is added or deleted the system sends appropriate Action and Attention items down- and up-stream in the system, requesting other users to the prepare or modify all of the training documents affected. New training documents can only be created by adding tasks to the Components cross reference system. Adding or deleting a task automatically sets Attention items throughout the entire system for all of the training documents that are required to be created or modified by the addition or deletion of a task.

CAPS Data Representation

The following paragraphs describe the components of the ARTEP database and all of its interrelationships as it is represented to the CAPS user. The reader should understand that this representation is a conceptual data structure and may not correspond with the way the data are actually stored in the computer system. This explanation explains data as they exist in the system, not how



PCT: 218 Camouflage Vehicle & Equipment
301 Perform Movement Security using Smoke
243 React to Indirect Fires
133 Recover your Tank

78 Reconnoiter an Obstacle
158 Defend Against Air Attack
147 Execute Actions on Contact
137 Perform Platoon Fire Distribution & Control
76 Execute a Hasty Attack
172 By pass an Enemy Position
169 Maintain Contact with the Enemy

FIGURE 49. Component Selection Dialog Menu

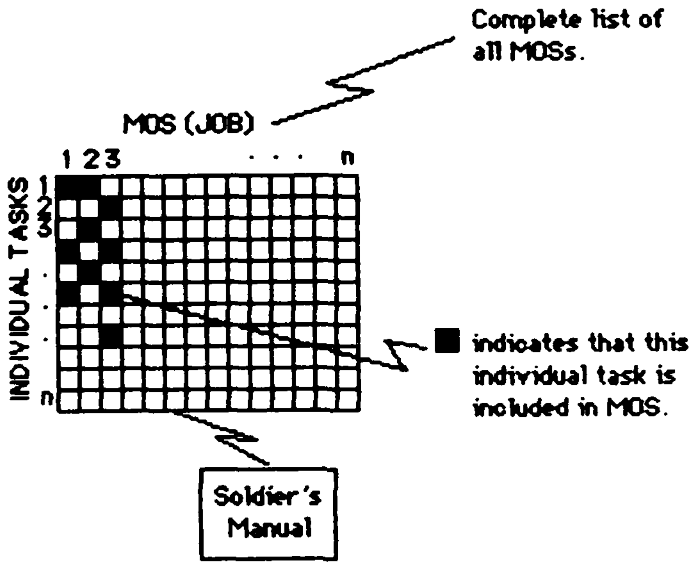
data are entered into the system. Note that some of the specific details of this data structure may not be completely accurate and will be modified in the implementation of the system. However, the data structure is sufficiently accurate to allow the reader to understand the interactive nature of this system.

Consider the MOS (Job) x Individual Tasks (MOSxIT) Matrix shown in Figure 50. The Individual Tasks dimension is a list of every soldier task described in the database. Obviously this is a very long list. Each item in this list corresponds to a description of a task identified in a individual Front-End Analysis or Soldier's Manual. The MOS dimension is a list of every job specialty included in the entire database. This is also a very long list. An entry in the matrix indicates that a given individual task is part of the MOS identified at the head of the column. This MOS x Individual Task matrix is an index/cross reference to every Soldier's Manual included in the database. The box below the matrix labeled Soldier's Manual indicates that there is a Soldier's Manual or part of a Soldier's Manual associated with every column in this matrix.

The Squad Collective Tasks x Individual Tasks/Squad Leader Tasks (SCTxIT) Matrix is shown in Figure 51. The Individual Tasks dimension of this matrix is the same list as the Individual Tasks dimension of the MOSxIT matrix, shown previously in Figure 50. The Squad Leader Tasks dimension of this matrix is a complete list of every Squad Leader task included in the database. This is also a very long list. The Squad Collective Tasks dimension is a list of every collective task performed by a Squad. Thus, the columns in the matrix each represent a different Squad Collective Task. An entry in the matrix indicates that the Squad Leader Task or Individual Soldier Task thus marked is included in the Squad Collective Task identified at the head of the column. The boxes below the matrix labeled Training and Evaluation Outline (T & EO) and Drill indicate that there is, or should be, a T & EO and a prescribed Drill described for every Squad Collective Task. These training documents are part of the database and this matrix is an index/cross reference to these training materials.

Consider the Squad Level Missions x Squad Collective Tasks Matrix, also shown in Figure 51. The Squad Collective Tasks dimension of this matrix is the same as the Squad Collective Tasks dimension of the SCTxIT Matrix. This is an exhaustive list of every Squad Collective Task. The Squad Level Missions is a complete list of every Squad Level Mission. An entry in the matrix indicates that the Squad Collective Task identified by the row is included in the Squad Level Mission identified by the column. The box below the matrix labeled STX (Situational Training Exercise) indicates that an STX could be defined for every Squad Level Mission. These STXs are included in the database and this matrix serves as an index/cross reference for all Squad Level missions and their associated training materials.

Next consider the Platoon Collective Tasks x Squad Collective Tasks/Platoon Leader Tasks (PCTxSCT) matrix, shown in Figure 52. The Squad Collective Tasks dimension of this matrix is the same as the Squad Collective Tasks dimension of the previously described matrices. Platoon Leader Tasks are added to this dimension. An entry in the matrix indicates that a given Squad Collective


Complete list of all MOSs.
MOS (JOB)
1 2 3 . . . n
INDIVIDUAL TASKS
1 2 3 . . . n
■ indicates that this individual task is inoluded in MOS.
Soldier's Manual

NOTE: Individual tasks can be grouped into squad collective tasks or into MOS catagories. These groupings may be different.

FIGURE 50. MOS (Job) x Individual (MOSxIT) Matrix

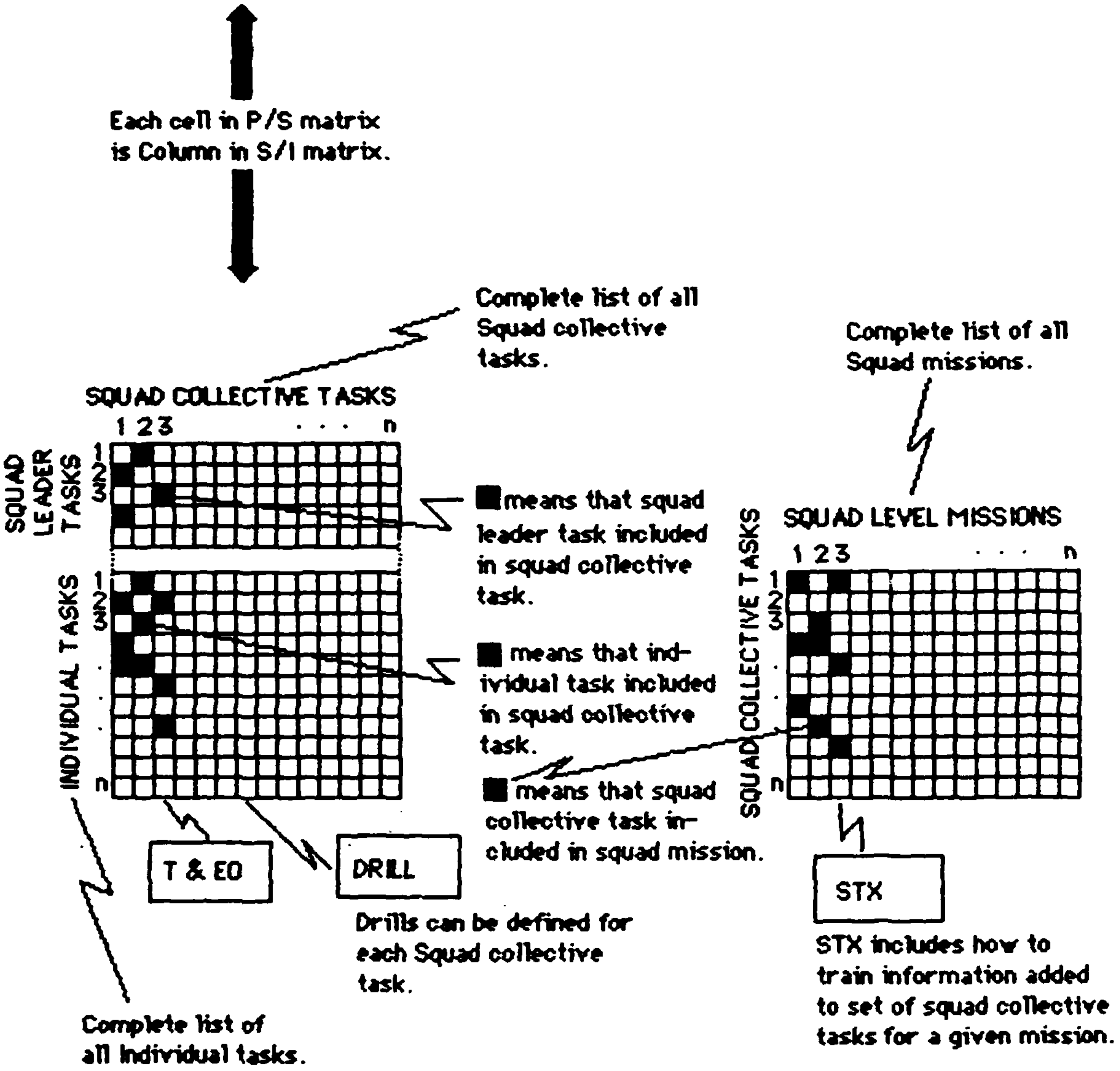

Each cell in P/S matrix is Column in S/I matrix.
Complete list of all Squad collective tasks.
Complete list of all Squad missions.
SQUAD COLLECTIVE TASKS
1 2 3 . . . n
SQUAD LEADER TASKS
1
2
3
■ means that squad leader task included in squad collective task.
INDIVIDUAL TASKS
1
2
3
n
■ means that individual task included in squad collective task.
SQUAD LEVEL MISSIONS
1 2 3 . . . n
SQUAD COLLECTIVE TASKS
1
2
3
n
■ means that squad collective task included in squad mission.
T & EO
DRILL
Drills can be defined for each Squad collective task.
STX
STX includes how to train information added to set of squad collective tasks for a given mission.
Complete list of all Individual tasks.

FIGURE 51. Soldier's Manual x Squad Collective Task x Individual Task (SMxSCTxIT) Matrix

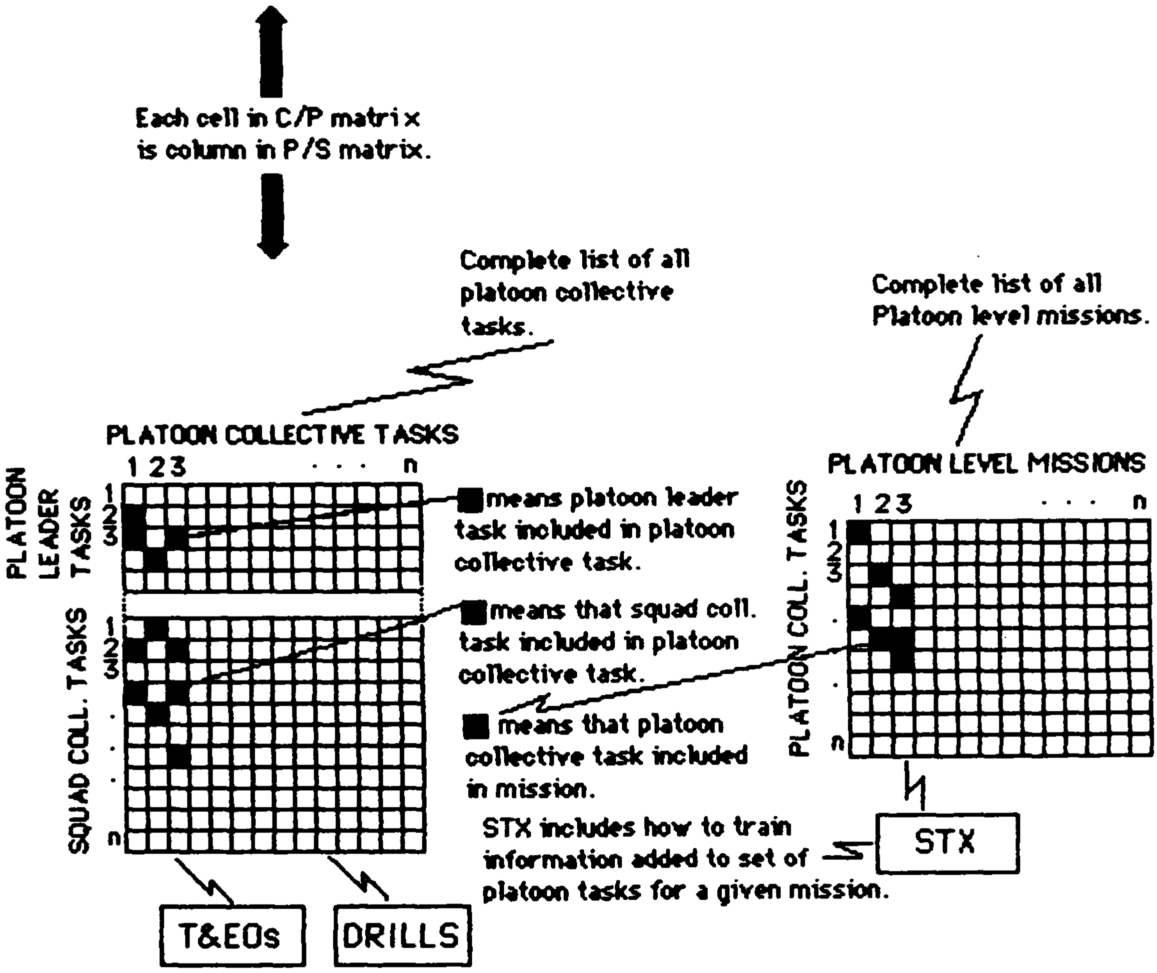


FIGURE 52. Platoon Mission x Platoon Collective Task x Squad Collective Tasks (PMxPCTxSCT) Matrix

Task or a given Platoon Leader Task is included in the Platoon Collective Task identified at the head of the column. The boxes labeled T&EO and DRILLS indicate that the database includes a T&EO and a prescription for a Drill for every Platoon Collective Task. This matrix serves as an index/cross reference for all Platoon Collective Tasks and the associated training materials. Note that each cell in this matrix consists of a Squad Collective Task column from the SCTxIT matrix, shown previously in Figure 51.

The Platoon Level Missions x Platoon Collective Tasks Matrix is also shown in Figure 52. The Platoon Collective Tasks dimension of this matrix is the same as the Platoon Collective Tasks dimension of the previously described matrix. An entry in this matrix indicates that the Platoon Collective Task identified by the row is required for the Platoon Level Mission identified by the column. The box labeled STX indicates that a situational training exercise can and should be described for every Platoon Level Mission identified by the matrix. This matrix serves as an index/cross reference for all Platoon Level Missions and their associated training materials.

The Company Collective Tasks x Platoon Collective Tasks/Command Tasks (CCTxPCT) Matrix is shown in Figure 53. The Platoon Collective Tasks dimension of this matrix is the same as the Platoon Collective Tasks dimension of the previously described matrices. Command Tasks are added to this dimension. An entry in the matrix indicates that a given Platoon Collective Task or a given Command Task is included in the Company Collective Task identified at the head of the column. This matrix serves as an index/cross reference for all Company Collective Tasks. Note that each cell in this matrix consists of a Platoon Collective Task column from the PCTxSCT matrix, shown previously in Figure 52.

The Company Level Missions x Company Collective Tasks Matrix is also shown n Figure 53. The Company Collective Tasks dimension of this matrix is the same as the Company Collective Tasks dimension of the previously described matrix. An entry in this matrix indicates that the Company Collective Task identified by the row is required for the Company Level Mission identified by the column. The box labeled FTX indicates that a Field Training Exercise can and should be described for every Company Level Mission identified by the matrix. This matrix serves as an index/cross reference for all Company Level Missions and their associated training materials.

The Battalion Collective Tasks x Company Collective Tasks/Command/Staff Tasks (BCTxCCT) Matrix is shown in Figure 54. The Company Collective Tasks dimension of this matrix is the same as the Company Collective Tasks dimension of the previously described matrices. Command/Staff Tasks are added to this dimension. An entry in the matrix indicates that a given Company Collective Task or a given Command/Staff Task is included in the Battalion Collective Task identified at the head of the column. This matrix serves as an index/cross reference for all Battalion Collective Tasks. Note that each cell in this matrix consists of a Company Collective Task column from the CCTxSCT matrix, shown previously in Figure 53.

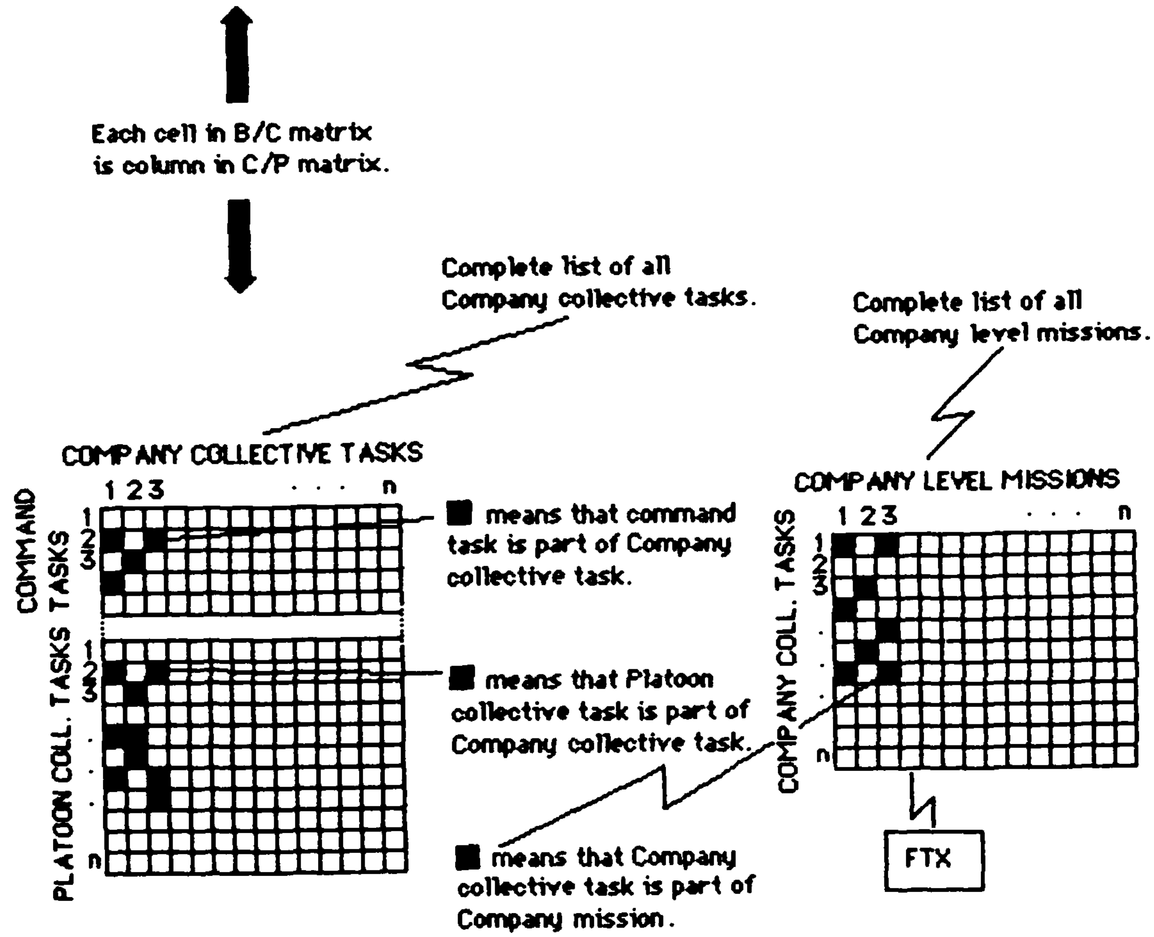


FIGURE 53. Company Mission x Company Collective Task x Platoon Collective Task (CMxCCTxPCT) Matrix

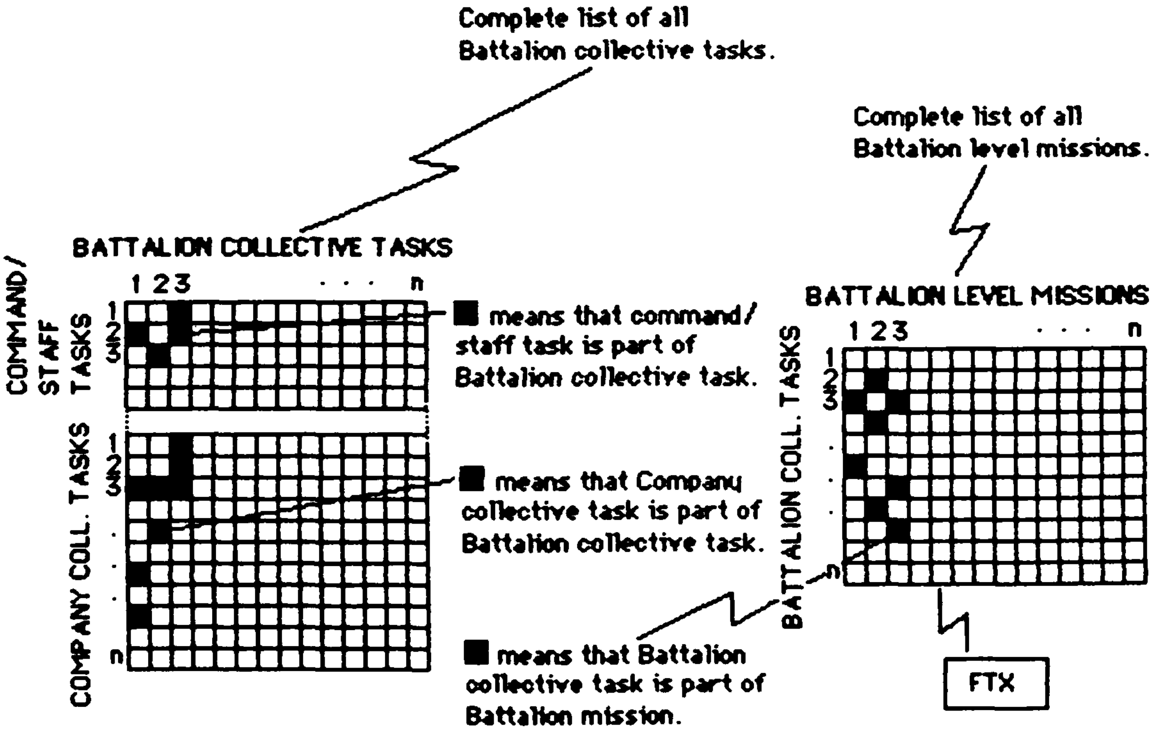

Complete list of all Battalion collective tasks.
Complete list of all Battalion level missions.
BATTALION COLLECTIVE TASKS
1 2 3 . . . n
COMMAND/ STAFF TASKS
1
2
3
■ means that command/ staff task is part of Battalion collective task.
COMPANY COLL. TASKS
1
2
3
.
.
.
n
■ means that Company collective task is part of Battalion collective task.
■ means that Battalion collective task is part of Battalion mission.
BATTALION LEVEL MISSIONS
1 2 3 . . . n
BATTALION COLL. TASKS
1
2
3
.
.
.
n
FTX

FIGURE 54. Battalion Mission x Battalion Collective Task x Company Collective Task (BMxBCTxCCT) Matrix

Consider the Battalion Level Missions x Battalion Collective Tasks Matrix shown in Figure 53. The Battalion Collective Tasks dimension of this matrix is the same as the Battalion Collective Tasks dimension of the previously described matrix. An entry in this matrix indicates that the Battalion Collective Task identified by the row is required for the Battalion Level Mission identified by the column. The box labeled FTX indicates that a Field Training Exercise can and should be described for every Battalion Level Mission identified by the matrix. This matrix serves as an index/cross reference for all Battalion Level Missions and their associated training materials.

Impact Relationships

Figure 55 is a diagram showing the various components of the ARTEP database. Please note that some of the specific details related to this iterative impact diagram may be inaccurate and will be modified in the implementation of the system. However, the diagram is sufficiently accurate to allow the reader to understand the interactive nature of the system.

This diagram explains the interactive and iterative nature of the data structure. However, the diagram does not attempt to indicate who makes the entries for a given part of the database or how the system alerts a given user that additional entries need to be made to keep the system up-to-date and complete. These concerns will be considered in the next section of this report. Since the system is iterative in nature, a complete description of all the possible interactions is impossible.

The components of the database have been divided into three types: 1) primary inputs, 2) training materials, and 3) cross reference matrices. The primary input components are indicated by the diagonal-strip boxes, the training materials are indicated by the grey boxes, and the cross reference matrices by the white boxes. The solid arrows show the primary impact paths while the dotted arrows show secondary impact paths. A primary impact path is one hat results from a change in one of the primary input components. This change requires updating of the components "down stream" from where the change was made. The priority system (described previously) would require, via the Action List, that the changes resulting from primary impact be made.

A secondary impact path is one that results from a change in the training materials that may suggest the need for the addition of an individual task, a leader or command task, or even a change in doctrine. Since secondary impacts move up the priority system, the implied changes in higher priority components can only be suggested (via the Attention List) and the actual change must be made by a user with the appropriate priority to make that change.

The system can be entered at any of the primary inputs (diagonal-strip boxes) or training inputs (grey boxes) indicated in Figure 55. The following descriptions indicate the iterations required from a change in a primary input component and also the iterations that may result from a change in a training input component.

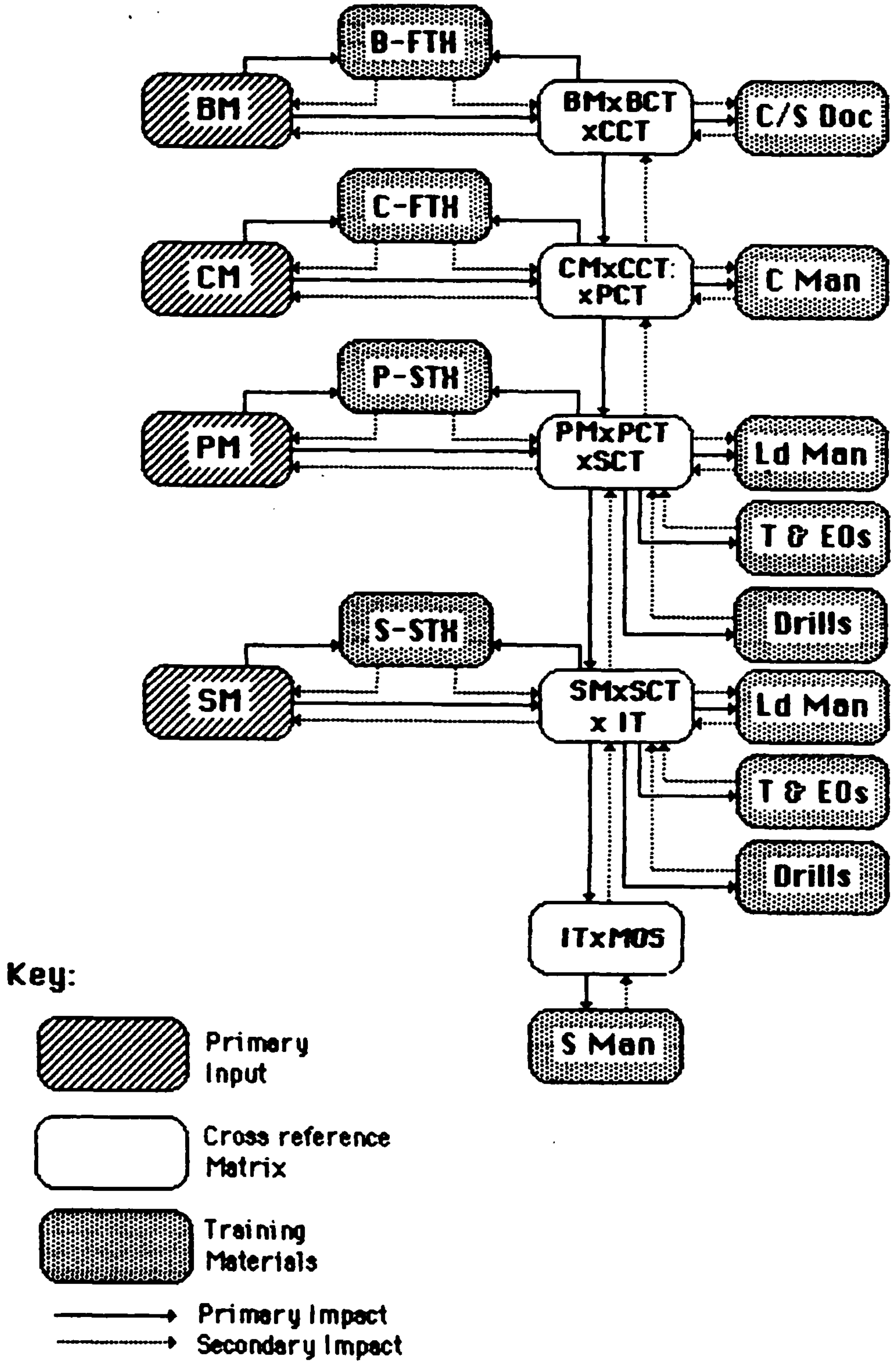


FIGURE 55. CAPS Iterative Impact Diagram

Primary Data Entry. A primary purpose of the ARTEP program is to provide adequate training materials to guide the execution of training exercises. The foundation of the program is the definition of training missions that prepare the fighting units to work together as a team in the implementation of the latest in strategy and equipment utilization. Changes in Doctrine and/or the equipment available to fight affect the way missions can be carried out and the strategies that might be appropriate. The tremendous value of the CAPS system is the ability for a change to iterate through the entire system, thus assuring that the training guidance provided is current and consistent with the most current doctrine.

Primary entry to the system can occur by defining training missions at any echelon. Missions for higher echelons have more potential impact than do changes in missions for lower echelons. To illustrate the iterative changes that this system makes possible, consider a change in a mission for the Company level. The user enters the CAPS system at component CM (Company Mission). The required entry is to define or redefine a Company level mission in accordance with a change in doctrine, strategy, or equipment availability. This mission description is part of the CAPS database.

A change in a mission statement iterates through the entire system as follows: A change in a Company level mission may require changes in one or more of the Platoon level missions that may be part of the larger mission. This impact is indicated in Figure 55 by the solid arrow from CM to PM. Recommendations for such changes are sent to the Platoon level mission component for action by the user who will modify platoon level missions. (This could be the same user, in which case his Action List would now contain the message to make the appropriate changes in the Platoon level missions.) The resulting change in a platoon level mission may require changes in one or more of the squad level missions which may be part of the platoon level mission. This impact is indicated in Figure 55 by the solid arrow from PM to SM. Action items initiated by any change remain in the system until the change has been made.

Returning to the initial change, a change in a Company level mission requires the user to update the Company Level Mission x Company Collective Tasks cross reference matrix. A change in a Company level mission may also require the addition, deletion, or modification of one or more Platoon Collective Tasks or Platoon Leader Tasks that comprise the Company Collective Tasks, which are required for the mission. In other words, a change may be required in the Company Collective Tasks x Platoon Collective Tasks/Platoon Leader Tasks matrix. This impact is indicated in Figure 55 by the solid arrow from CM to CMxCCTxPCT. The system prompts for the necessary changes to the matrix.

Changes in the CMxCCTxPCT matrix may iterate throughout the entire system. For example, a change in a Platoon Collective Task may require the addition, deletion, or modification of one or more Squad Collective Tasks which comprise the Platoon Collective task. This would require a change in the Platoon Collective Tasks x Squad Collective Tasks/Squad Leader Tasks matrix. This impact is shown in Figure 55 by the solid arrow from CMxCCTxPCT to PMxPCTxSCT. A change in a Squad Collective Task may require the addition, deletion, or modification of one or more Individual Tasks which comprise the Squad Collective Task. This would require a change in the Squad Collective x

Individual Task matrix. This impact is shown in Figure 55 by the solid arrow from PMxPCTxSCT to SMxSCTxIT. Furthermore, a change in an Individual Task may require the modification of an MOS and resulting changes in the Soldier's Manual which describes this MOS. These impacts are shown in Figure 55 by solid arrows from SMxSCTxIT to MOSxIT to Soldier's Manuals.

Other iterations are also possible from the change in a Company level mission. Some of these are as follows: Changes in the PMxPCTxSCT matrix may imply changes in platoon level missions. This impact is a secondary impact shown by the dotted arrow from PMxPCTxSCT to PM. A similar secondary impact on squad level missions may also result from subsequent changes in the SMxSCTxIT matrix as shown by the dotted arrow from SMxSCTxIT to SM.

The end result of a mission change is the necessary changes in the training materials. A change in a Company level mission would have a direct impact on the preparation of Company level Field Training Exercises, as shown by the arrow from CM to C-FTX. The ripple effect described in the previous paragraphs may result in changes to Platoon Level Situational Training Exercises, indicated by the solid arrow from PMxPCTxSCT to P-STX, and in Squad Level Situational Training Exercises, indicated by the solid arrow from SMxSCTxIT to S-STX.

Other training impacts are changes required in T & EOs and Drills by changes in the PMxPCTxSCT matrix and the SMxSCTxIT matrix. Changes in the Soldier's Manuals resulting from changes of individual tasks have already been indicated.

A careful study of Figure 55 shows that a change at a higher level may iterate throughout the entire system. The great advantage of the CAPS system is that each of these impacts is systematic, is made part of an Action List for an appropriate user, and remains as a flag in the system until an appropriate change has been made.

Secondary Impacts. Secondary impacts can be side effects of primary changes as indicated above. The other source of secondary impacts comes as a result of insights gained or inconsistencies observed when a given training exercise or manual is being prepared. The message concerning this inconsistency ripples "up stream" through the system, setting messages on upstream user's Attention List. For example, when writing a Soldier's Manual, a user may realize that further action is required farther "upstream" and may message the Attention List of a user at PM via the PMxPCTxSCT matrix.

A complete tracing of the iterations caused by a single change are difficult or impossible to trace completely. The previous discussion should be sufficient to assist the reader in understanding the iterative nature of the system and how the impacts ensure that a change in one part of the system results in the necessary and potential changes to other parts of the system. The result is training materials that are always current with the latest in doctrine.

CAPS Database Management System

General Motivation. The preparation of an MTP requires the interaction of the author(s) with a large and complex variety of data. These data represents the accumulated relevant reference information from other Mission Training Plans, basic reference sources (Soldier's Manuals, standards, etc.), the work product for any previous versions of an MTP undergoing development, along with the current status of the MTP being prepared. Indeed it is likely that more than one version of an MTP (or parts of an MTP) must be maintained in the system and that the system undergoing development has several different versions (or levels of currency).

The system itself must be able to provide support for the management of the diverse versions of documents we have mentioned. It must form an archival base for an MTP once frozen. It also should be capable of imposing structures so that we can authorize changes to various parts on the basis of responsibility, to eliminate the possibility of even inadvertent changes made by non-authorized personnel.

We also recognize that a wide variety of auxiliary information relating to the production process itself will be managed by the underlying computer-based system. Such things as the storage and retrieval of documents relating to the work in progress, and a complete tracking system to manage internal distribution of drafts, routing of information, and the status of signoffs will be handled.

The best means of accomplishing the task of *managing this information* is building a superstructure on top of a commercially available, large-scale database management system. By doing so, we can leverage off many of the underlying features of such a system, including the physical management of the data and the ability to perform queries (both standardized ones that are systematically applied to the data to check for consistency, etc. as well as ad hoc queries generated interactively during MTP development). Such a system typically satisfies our needs in the area of data access protection through the use of user names and passwords that will permit and restrict accessor modification of fields as desired. Modern database management systems also provide a wide variety of library and language access methods. This will permit us to integrate tools and programs from a variety of sources and disciplines to access and manipulate the data without being restricted to one particular language, which would typically occur if the database storage were specifically done for CAPS itself.

Requirements. As a result of our studies of existing Mission Training Plans, we see that the information is more than standard textual material. it appears that about 50% of the material is textual and the remainder is either graphic images in the form of figures or tables and charts. This graphical information, as well as the textual information, is highly variable in length and content. It is not at all like the simple data or other disciplines. Hence, we need to maintain, manipulate, cross reference, and correlate significantly mixed data representations demanding a highly flexible system.

The prime requirement for storage of the task description information is the ability to handle data structures which contain perhaps several thousand rows. Each row contains a relationship between a particular element in a Drill or STX, etc., and the other elements, such as necessary preconditions, corresponding actions by related groups, references to doctrine, etc.

For example, let us consider the information needed to describe a particular ARTEP Task in a form that would be used in preparing the MTP material. Such an element would comprise at least the following bits of information:

(1) Drill-id: This is the internal reference number (assigned by the system in a unique fashion) that is used to relate together different elements of an MTP. This number is not used for any external purposes; for these purposes see data elements ARTEP-Task-No or T&EO-Task-No.

This field is a fixed length field containing a large integer or string.

(2) Description: A particular task has an English language description that categorizes this item in a highly syntatic form; e.g., FORM COLUMN, TAKE ACTION ON CONTACT, etc. Note that the words and phrases used here come from a small vocabulary and we expect that CAPS will include input validity checking as well as analysis of completeness available to check the completeness of the MTP.

Note that this field might be several words long and of highly variable length.

(3) Echelon: Each task applies only to a particular individual or collective level. This field is chosen from the small set of possibilities; i.e., Battalion, Company, Platoon, Squad.

This field could be compressed into only one byte.

(4) Form of Task: Tasks are referred to by different terminologies depending on their context. This field contains the official term to be used, such as TASK, DRILL, STX, FTX, etc. Note that although these are different types of elements, they share so many other things in common (such as the preceeding fields, etc.) that for the purposes of the internal computer database we are not making a formal distinction among them, even though one exists. They are being grouped together because that simplifies the handling by the computer system since they share the same processing and storage requirements.

This field is also one of a few special types and could be an internal number or else a small character string.

(5) Reference: Every task is not fully defined in the body of the MTP itself but often the details are found in other material such as the Soldier's Manuals. Hence, we need a field (or several elements) that are these reference pointers. They will contain standard reference labels, such as FM-17-19E1/2, which will be printed in the MTP and looked up by instructors. One form of cross checking that CAPS should make possible would be to see the completeness of MTPs; are all relevant portions of a Field Manual included or not? Technically this is part of the Collective Front End Analysis and not the actual authoring of the MTP. Nonetheless, it is essential that the database permit such an extension of its usage.

This field is a variable length string; it might be tens or hundreds of characters long.

(6) ARTEP-Task/Mission: Every task belongs to one of the generic set of ARTEP classes; for example, it might be MOVE, DEFEND, or ATTACK. (There are also subcategories, such as ATTACK-HASTY, or ATTACK-URBAN.) We might create a second field for the ARTEP-Subcategory; e.g., HASTY, URBAN, DELIBERATE, etc.

This field can only be filled with one of a small set of legitimate terminology. Hence, it could be reduced to small internal numbers, or else a small set of possible strings.

(7) ARTEP-Task-No: This field will contain the present ARTEP task number (where appropriate) or else other existing relevant designators, such as the Soldier's Manual or Field Manual number. This could be something such as 3-IV-1-5 or 171-127-0002.

This field would be a variable length string several tens of characters long.

(8) T&EO-Task-No: This field would contain the present index for an item, such as Task 147, etc. This field is mostly used for cross referencing when showing STX-FTX matrices, etc.

This field is typically an integer.

(9) Decay-Factor: Every task (in general) has a decay factor associated with the frequency of refreshor training needed. For the instructor to determine whether sufficient drill and practice is being given, this decay factor has to be calculated in combination with other decay factors that result from training in various different contexts. This is another area where the database would be useful in an active fashion, going far beyond the production of the MTPs, but in their usage by instructors. Using this information, a set of preparations could be evaluated to see if they are sufficient in training various areas, over time.

This field would be a number relating to the time factor.

(10) Other Fields: It is hard to say at this time all the relevant fields that we might develop from a complete and systematic analysis of a given MTP. It is fair to say that additional fields would be useful and there might be several such, each multiple variable character strings long.

Thus, we get a picture of a standard record entry for each kind of TASK; it is comprised of up to several hundred bytes of data. We see that there is a lot of variability in the actual amount of data in use in each particular record. Some will have lots of entries in the reference fields, and others mostly in the "other" category. Hence, it is a requirement that the system permit records that are hundreds of bytes long (and shortly we will see why they must be in fact thousands) and also that the system compress out all the unused space, or else we will require two or three times the raw storage.

We have made an estimate of the number of entries in a table such this one, and it is several thousand records long.

If we proceed to study the data requirements for descriptions of missions and drills, we see other types of data elements are required that are not addressed above. They include both graphics information (see Section 4.1.6) and lengthy textual descriptions. Both of these data types are represented by very long, variable length strings of data. In the case of pictures, the description of a picture is composed of individual elements of primitive form, such as rectangles, circles, connections, shading, etc. Each of these elements is qualified by the particular size, shape, position, etc. This requires many thousands of bytes to represent a typical tree of tasks or other objects such as formations, signals, etc. Similarly, for text we see the need for sections of text that are as long as a page (or so); this is also as much as several thousand characters, potentially. Hence, we conclude that one requirement for the database system is the ability to manage files of up to several thousand records, each one potentially variable in width from tens of characters to tens of thousands of characters long.

We also recognize that we cannot, at this time, define all the possible uses of the data, their possible values, nor all the fields that might be appropriate to add to the data once we have them in manageable form. Hence, it is a requirement that there needs to be an ability to add additional fields to a record, to adjust the size of a field, and to generally restructure the database without recreating it all, during the lifetime of these data.

As well as recognizing that we cannot anticipate all the possible fields of data that the system will evolve to require, we expect this is also true in the area of database inquiries. The graphical front end will provide for many standard options for presenting and visualizing the current status of the plan being developed, but it is clear that there will be many additional ad hoc queries that we cannot even anticipate at this time. This is most certainly a strong requirement during the initial development of the system. It will become less critical as the system is rolled out in operational use, but we believe that because of the radically new set of tools that will evolve only during exposure to the system. Hence, it is a requirement that the underlying database management system must have a high level (English-like) query interface. As well, we believe that the interface must be available from a set of languages, to include at least the following: assembler, C, and Pascal.

The design of the system architecture is a series of intelligent front end graphics-oriented workstations, with a back end database server/machine. This structure imposes an additional requirement on the database management system; namely, that the DBMS must be networkable. A suitable DBMS must be able to be split into multiple tasks, running on different physical machines, which are communicating via a high-speed network. We envision this split providing for the ability of the user stations supplying fair high-level queries (those described in the previous paragraph), which are sent to the back-end processor. This back-end processor has considerably higher performance than the workstations and, as well, is the repository of the actual database in disk form. This processor is the place where the locking of the records takes place during updates, and is used to coordinate the demands of multiple users. Any query will cause activity to take place on this back end machine with the results (generally now a small subset of the database) to be sent via the high-speed network to the individual work stations. During this transfer, the data may be buffered using some intermediate node file storage, although from the logical point of view this is totally transparent.

Our analysis of the nature of the changes to the database for the CAPS system shows that the development of an MTP is an iterative process that involves a certain amount of "trial and error." That is to say, an author develops one approach, then analyses it to see if it meets the requirements and coverage of fundamentals. This may result in another attempt at a solution, and so on for several cycles. During this process, many related changes are made to the database, which must all be backed out to enter a new approach. It is a requirement that the database system provide "rollback" to a previous status as a standard underlying feature. As a related matter, the system must provide a "commit" facility, which makes the set of changes official in the database and available to other users.

Hierarchical Database Systems. We may be able to best understand the nature of a traditional hierarchical file system by way of an example (taken from Date). Imagine that we are keeping track of a series of parts that we receive from several different suppliers. A given part may come from different suppliers, and a given supplier may supply several parts. One way to organize these data might be to first consider all the different parts. For each part we can list the part number, part name, part color, part weight, and location of the part. In addition, for each of these records we must keep track, by means of a pointer or a set of pointers linked together, the supplier for that part, location of that supplier, quantity of parts we have in stock from that particular supplier, etc.

The typical nature of queries made against such a database would be: give me the first part; give me the name of the current part; give me the next part; give me the supplier number of the current part (or the supplier record of that supplier number). To ask a question such as, "Show me all the part numbers of all parts that all the suppliers in Buffalo provide," the data must be found by first studying the explicit links that connect together the individual records. In the case described above, we presumably would search through each of the parts, looking at each of their suppliers, and if the location of the supplier was relevant, keep track of the original part number.

On the other hand, one could have organized the data in this example by making a list of all the suppliers, and then for each of the suppliers keeping a list of pointers to each of the parts they have supplied. If this was the case, the program that was responsible for supplying the answer to the query we are considering would be different. It would immediately look at the location of the supplier and only then, if in the correct city, make a list of the part numbers.

This is not a good state of affairs unless we have a very strong reason to prefer one organization over the other. Hierarchical systems force all the programs that access the data to know quite a bit about the organization of those data in the forms of the trees or links between records. It is possible to apply many well-known techniques of indexing and hashing to make several different alternative routes through the data relatively more efficient, but the model of the sequence of records, subrecords, detail records, etc., permeates the hierarchical system.

Another problem to which most hierarchical systems do not lend themselves to easy solution is the elimination of redundant information. In the example above, once we have defined the structure, if we choose to keep the supplier location in the supplier record, then the first method results in many copies of this information. When the supplier moves to a new city, the change must be made in many places. It is possible to keep a separate record for each supplier and look up the location there, but this is no longer a strictly hierarchical tree system. Instead, if we try to make this change once programs have been written to each datum (or to suddenly add multiple locations to a given vendor), it becomes necessary to find and modify all the relevant programs.

In addition to the lack of symmetry in hierarchical database systems and the problems associated with avoiding redundant information, experience has shown that making changes (e.g., insertions, deletions, new record types, increasing the size, squeezing out the "dead space," reorganizing the data for more efficient searches, etc,) are extremely difficult to do well, especially when the timing of making some of these changes must be synchronized with simultaneous program changes.

Yet hierarchical systems are very natural models to use for data that arise from the "real world." Mission Training Plans are organized into volumes at levels Battalion, Company, Platoon, etc. Then each of these is organized into chapters, and so on. This is the natural way that we organize information in our physical filing cabinets and is the way we tend to initially present the printed version of the Training Plans. Hence, there is a strong natural tendency to view the data that this particular database (the CAPS Database) must handle as strongly hierarchical in nature.

Relational Database Systems. Relational database systems provide for a great deal of shielding of the internal details of the management of the actual data and allow the programs to deal with the data and data relationships from a more abstract position. Let us reconsider the simple example of the last section from the perspective of a relational system.

In a relational database system, we refer to a collection of records as a two-dimensional table, known as a "relation." A collection of several tables typically comprises one "file." Each of the "rows" of the table corresponds to the record that we discussed in the previous section. The only relationship between data that appears in different tables is through values that are in common between these tables; this might be elements such as "part number," "supplier name," etc.

A simple relational database, consisting of three relations, is shown in Figure 56. In this case, we would probably organize our example by providing one table for each vendor number, name, and city. (Note there is no restriction that there must be only one entry for a given vendor number; hence, multiple cities might be allowed. Of course, we could also disallow such a construct if that is never possible.) Notice that the name and location of a vendor (potentially long address strings which occupy lots of disk space) appears only once, thereby occupying minimal storage and alterable in one location. A separate table would list the part number and part name.

Finally, we would build a third table that has entries with three elements: the part number, vendor number, and on-hand quantity that corresponds to that pair. Notice that all the links between the first two tables is done via the third. But a relational system does not even require the user to be aware of that matter. It will be able to provide the answer to queries such as "list the parts that are supplied by suppliers in Buffalo" by having the database system handle the necessary linking of the information. Of course, it is still possible to make significant differences in the amount of time it takes to answer such a query by reorganizing the data or its sequencing. In this case, the linking table could be organized along the sequence of supplier number, or part numbers, or perhaps other techniques (hashing) that handle both might be selected. But in a relational system such choices do not affect the ability to get results or the programs that access such results. They do change the speed by which the results become available.

The Choice of a Relational Model for CAPS. The analysis of the ARTEP training material has indicated the great amount of structure contained in the information. There is also a large amount of parallelism between the information and organization at various levels or echelons. In fact, it is essential for the authors to maintain this parallelism, since activities at each level must be coordinated. While we are not saying that higher-level tasks are simply aggregates of lower-level ones, they certainly require or imply certain actions at lower levels. Squad activities cannot take place without individual soldier action, etc.

Today's training materials present a view of the underlying CAPS data according to certain traditional methods. The charts that display the tasks which comprise an STX are found to be effective in communicating that

Vendor No.	Vendor Name	City	State
045981	Good Wines, Ltd.	San Francisco	CA
337980	Little Computers	Manassas	VA
125065	AeroSpace Company	Los Angeles	CA
887320	Rags & Such	New York	NY

Vendor Relation

Part No.	Part Name
C667	Floppy Disk
B872	Ribbon Cable
H90x	Print Head

Part Relation

Vendor No.	Part No.	Quantity
097564	C667	100
337980	A012	440
125065	A075	25
337980	H90x	100
045981	A012	123
887320	C667	224

Vendor-Part Relation

FIGURE 56. A Small Relational Database

information. STX-to-FTX matrices are another tool that is useful in structuring exercises and training, but are really just another way of looking at information that is already contained elsewhere in the MTPs. In fact, we know that it is probable that despite efforts at correctness, these two ways of presenting the relationship may have errors or oversights that we would like to eliminate. A relational database provides the natural mechanism to deal with such problems. The data themselves are entered in only one form. Any change to that information will automatically propagate itself to the different ways that this information is used by authors and in the actual manuals themselves. As well, we can more easily limit the authorization to change information in ways that model these authorities.

Let us look at the example we have been using to understand this point. In a relational model, we can give permission to modify the part number table to one individual, permission to modify or add to the supplier table to a second individual, and the third table which describes the actual inventory of parts in stock to a person with different responsibility or authority. Trying to do this in a hierarchical system is not very practical.

In the case of the data needed for the CAPS system, there are great advantages derived from allowing the links between data to arise from the data themselves. This will permit us to load up the database starting from "raw" documents, and then gradually produce simpler and simpler tables from the data. By way of this example, we might initially enter the information by explicitly naming each supplier, supplier location, etc. Then, using the ad hoc query access of a relational DBMS, we could request things such as "a list of all supplier names, sorted." A quick scan over this list will quickly identify things such as misspellings. Once this sort of thing is cleaned up, we can build a table of supplier information, etc. As well, we can see that a relational database system makes it quite straightforward to add additional fields to existing data. This is far from simple for a hierarchical model in most typical systems.

Comparison between Oracle and Ingres for use as CAPS Database System

A detailed study has been made of the features of the Oracle database system, and the Ingres database system, to determine whether one was strongly preferred over the other, or not. Given the requirements of the CAPS database (as described elsewhere in this report), it is concluded that Oracle is preferrable over Ingres; however, Ingres is marginally acceptable. Oracle provides significant superiority in the following areas:

(1) Number of elements and total record size permitted
(2) Miscellaneous features

The following is an elaboration of the features that are of significant advantage to CAPS; most others are sufficiently close or irrelevant to CAPS that they are not mentioned. For example, the price of the two systems for use on a multiuser minisystem (such as a VAX 11/750) are quite similar. Report writing capabilities, and the ability to access from high level languages are also both quite sufficient for CAPS purposes.

Number of Elements and Total Record Size Permitted. Each record of an Ingres database is composed of up to 127 columns. Each column is a particular datatype, numeric or character. Numeric fields take up fixed length (1, 2 or 4 byte integers, 4 or 8 byte floating point numbers); character strings may be as long as 2,000 characters long.

The total length of all fields together must add up to no more than the 2,000 character maximum, and this is true despite whether the fields are actually present or have NULL values. Ingres does not compress space out of unused fields (such as numerics) unless specifically set up to do this, and will remove only trailing blanks in character fields.

The microcomputer version of Ingres, MicroIngres, (which might be used on the front end 68000-based workstations) is even more restrictive on record sizes; it restricts the number of columns to 49 instead of 127, and the rows to 1,006 characters wide instead of 2,000. It should be noted that if the database code is to be executed locally for certain operations, the database that runs in the central host would then also need to be limited to these latter numbers; which is an unacceptable limitation for CAPS.

Oracle permits each table to have up to 254 columns. Each element may be of type DATE (7 bytes long), NUMBER (m,n) (variable length number, up to m decimals digits long, with n decimal places), CHAR(n) (character string up to maximum length 240 characters), and one entry may be of type LONG, which is a variable length string up to 65,500 bytes long. There is no fixed maximum based on the sum of the lengths; it is legal to have 100 fields each of 200 characters if desired.

The Oracle database system uses no file storage for missing fields, hence it can compact data far better than Ingres. The microcomputer version of Oracle is fully compatible with the minicomputer version, and is not a subset.

Comparing these two alternatives, one may conclude that Oracle is significantly easier to use for the CAPS project. The most significant area of distinction derives from the 2000 character per row limit in Ingres (even less in MicroIngres). In CAPS, it will be essential to keep large (especially free-form text or graphics) objects in the database. It is possible to do this using Ingres, by using host operating system file names, and being very careful about lengths of fields and records, but a high price is paid in complexity to do so. As well, the lack of complete compatibility between mini and micro versions is not attractive.

Built-In Multiuser Design Features, Access and Concurrency Control

Both Ingres/Quel and Oracle have a large collection of features designed for a multiuser database environment. Users are known by their log-in names, and must provide identifying passwords. Based on the user identification, both systems will permit access to certain data, keeps track of ownership (or creators) or data, access to data, or subsets for purposes of reading, updating, etc. can be authorized on a user by user basis. All attempts to access data that is rejected by the system can be logged in an audit trail.

Both systems understand the requirements for concurrency control. The Oracle mechanisms are more highly elaborated than those of Ingres. But in general, in both systems, users may perform database locking on a record or table wide basis. Updates to data may be temporarily not committed to the database so that a partial, inconsistent view is not seen by other users. As well, experimental changes may be backed out.

Further details on the additional mechanisms that will be incorporated in CAPS are presented in the following section (Protection Control and Authority in the CAPS System).

Miscellaneous features. The documentation of Ingres appears to be directed towards a far more computer systems oriented user. It has far lower quality and readability than the corresponding Oracle documentation.

Oracle also has a convenient feature for turning relational data back into a standard hierarchical (or tree) oriented format. As discussed in the section on the choice of CAPS database system, much of the data we are dealing with does indeed have a tendency to be expressed conventionally in a nested or tree structure. This is true for much of the mission planning information and is certainly the case for grap'ics images, which contain subimages, etc.

In Oracle you can automatically take a database which is stored internally as a relational database and cause it to be processed or displayed in its natural hierarchical format. Consider, for example, a database which contains employee records and, for each employee, contains the id-number of his manager. This can be maintained as a simple, flat set of records in a relational system. But in Oracle, by using the CONNECT BY verb, you can identify the manager-of field, as really the way to find the id-number of the employee who is the manager. Then, by using the START WITH verb, indicating we would like to start processing with, say, the President or a Department Head, the system will provide records according to the natural tree order in the data. In the case of Ingres, this takes explicit programming to do a very common task.

Lastly, we mention a small but powerful feature that Oracle possesses that Ingres omits which we feel will greatly help our interactive users and during initial database clean-up and loading. In Oracle, there is a SOUNDEX function, which produces a string that tends to be identical despite misspellings in names or words. Using this field in the WHERE clauses, searches can be used to find data elements that one believes are in the database but may be misspelled.

Protection Control and Authority in the CAPS System

This section addresses the access controls in the CAPS system. To do so, we describe it from the point of view of the underlying database system. These are the inherent capabilities of the system, as unmodified by intervening front end programs. These front end programs, running on the intelligent workstation front end machines, can superimpose yet another view of the system on top of the one of the underlying database. This is one great advantage of using workstations rather than terminals; we can for example, impose additional security measures in front of the Oracle system, such as further checking of identity, further logging of information and queries to audit trails or journal logs, without having to actually modify Oracle itself. We will not describe these features here since we do not have any detailed improvements needed at this time beyond the Oracle ones. They may arise from possibly simple but different military operational requirements, such as standards for lengths and types of passwords, and frequency of change.

All users are identified to the system by their name (a variable length string) and password associated with that name. Without the password a user cannot get access to any of the data or rights to data associated with that name.

The creator of a table generally has the right to modify and manipulate all the data in the table. The owner can selectively GRANT some subset of rights to another individual, and can later decide to REVOTE all or some of those rights. (It is possible to permit the person to whom rights are granted the ability to further pass them on by himself, using a WITH GRANT OPTION, but we do not expect to utilize this feature for CAPS.)

The set of rights that can be controlled for each separate table in the database are:

a. SELECT. The ability to reach data, and to select some subset of that data, based on logical conditions, for display. By permitting someone SELECT access you are enabling them to know of the existence and contents of the data. (But see below for ways to do this on a defined subset of the data, including dynamically.)

b. UPDATE. This is the ability to modify some values, e.g., columns of the SELECTed data. Without this right, someone might be able to display information, but not to modify it, intentionally or accidentally. We expect that almost all data in CAPS will be able to be read by anyone authorized to use CAPS, but that in general, only one individual will ever be authorized to modify it. This could be extended to be a small set of separate individuals in the same team though, which CAPS will provide for.

c. INSERT. Inserting a new record in the ability to add a new entry to the database. Generally this will be done only by the person who has the privilege to UPDATE the data.

d. ALTER. This right permits a user to add a new field to the records. This is highly undesirable, except for the initial creator of the system, and will probably be restricted to use by the Data Base Administrator, once the system is stable.

e. INDEX. This permits the creation of auxiliary indices which speed up databases searches. This will be a controlled function, performed by the Data Base Administrator, since it can be extremely time and disk consuming if used incorrectly.

f. CLUSTER. This is also a space and performance related option, which we will likely restrict to use by the Administrator.

The names of tables in the multiuser environment are always qualified by their owner, e.g., if FRED is the creator of the table EMPLOYEE, then everyone (other than Fred) must refer to it as FRED.EMPLOYEE. This is done to permit many people to select identical names for their own tables, without restriction. However we intend to make heavy use of the SYNONYM feature of Oracle, which will permit us to use a global name (like EMPLOYEE) which will automatically refer to the appropriate specific table.

This is exactly the mechanism that will be used to deal with the on-line maintenance of versions. Each version of the databae will be periodically frozen, either for external distribution or for internal purposes. Current versions will be the default tables referred to. There will be a standard naming convention to get to other, non-current versions. For example, one might use EMPLOYEE for the current data, but use JUME84.EMPLOYEE to get the previous version. Note that if there have been no changes then we can further use the synonym mechanism to avoid making additional copies of things which do not change.

One particular power feature of a relational multiuser database is that we can define VIEWs of data, and give rights to those views only. For example, we might define a VIEW of data to be those employees in Department 123 only. By giving both SELECT and UPDATE rights to a user for this view, this user may see only a subset of the original complete data, and he is free to date that subset. He cannot tell that there is additional information in any way at all.

More interestingly, one can build fairly complex dynamic views. For instance, a view might be defined in terms of certain properties of the current user of the database. This might be something which permitted any user to see just those elements of the Employee records for all people that are in the Department of which he is the manager. This view can be defined once, and then used by anyone at all. Only people who are managers will be able to see any data at all. Note that we have avoided defining new views every time we add a new department.

One way we can use this ability is to permit individuals managing the CAPS database to update ony those fields that they have created. Or else, the access rights can be controlled by some other set of data tables which are more complex than the access rights we defined earlier. We might permit people in particular departments to update any data created by people in the same department. Oracle's views can handle such a situation, through the use of an auxilary table.

Oracle will automatically log any access violations (that is, attempts which were rejected) into an audit trail. This will ɔe used to monitor the system. As well, Oracle can keep journals, which can keυp track of all changes to the database, so as to permit us to recreate it serially should some disaster strike. These journals are normally written (as are the audit trail) to a tape, so they can be recovered most easily. We wish to keep track of all changes made to the system on a user basis as well.

The complexity of the CAPS database and modifications to it, make it important that a series of related changes be made to the database, and then installed all at once. If this is not done, then it is possible that other simultaneous users of the system will see partial changes only, and their own modifications, or reports, etc. will be inconsistent or incorrect. Oracle permits us to address this via the COMMIT facility. A series of database blocks are set aside for "partial" changes. We can turn the standard AUTOCOMMIT feature off. Once this is done, all changes to the database (in the form of UPDATEs, DELETEs or ADDs) will not be done at once, that is, they will be kept private to the user, who will see them, but no one else in the system will. Then later, only after all the changes are made, or if it is decided to undo the changes, the user can issue a COMMIT WORK command, which will make the changes take place, or else a ROLLBACK WORK verb, which automatically does all the intervening changes.

Lastly, the system will provide for the ability for multiple individuals to work on the same database, or tables, without destroying data by attempting to simultaneously update the same information. The automatic way that data is handled is to lock out any records which are being selected for update, so that no other use may access them until one user is completed. In fact, the underlying system will either automatically wait and retry later, or we may have it return to the front-end software with an indication to the user that is is "in use" temporarily. The Oracle system also detects and prevents any deadlocks from occurring, so that users cannot wait forever.

It is impossible to implement even more complex sharing scenarios using Oracle primitives. One can explicitly LOCK a table in SHARE MODE, which permits other users to read the table, but prevents them from doing any other updates until the entire table is explicitly released by the locking user. The table may be LOCKed in EXCLUSIVE mode, which means that any user which attempts to update any portion of the table causes the entire table to be locked until the update is complete. A more flexible approach is the SHARED UPDATE MODE, which permits multiple people to access different records, with those records being locked out from other users whenever they are being updated. At this time we do not envisage using the first two of these modes, but do expect to make the third be the default mechanism, to prevent any possible confusion during the time the database is being updated by any CAPS user.

CAPS Information Processing Design

System Architecture. The architecture that best suits the development and long range growth variability (from site to site) is a central host, responsible for data storage and management, supported by a set of highly intelligent, graphically oriented workstations, connected via a high speed network.

The central CPU will be used for all software development, bulk entry of data from existing ARTEP documents, the physical production of masters of the MTPs (for printing via standard presses), and as a database engine that serves the ARTEP writer's workstations. This CPU will also be responsible for routing files from site to site once there are several of the CAPS system deployed.

We have considered using highly graphical oriented terminals connected to the host CPU rather than intelligent terminals. But our analysis of the amount of central processing power needed to provide the responsiveness that users now expect from such personal computer workstations as the IBM PC/AT, or the Apple Macintosh, is such that it is far more inexpensive for all but the very smallest configurations to use intelligent workstations.

The following illustration, Figure 57, shows the general organization that meets this requirement. We now proceed to elaborate in detail the requirements for each portion of the system we have described.

Host Computer Requirements. The central CPU will be used for all the main storage requirements of the system. On this machine we find the major peripherals of the system. They include the master disk units, tape input and backup devices, hard copy master production device, optical character reading stations, communications links between sites, and general purpose terminals for program development.

Since the main application determines the responsiveness of the system, and hence the productivity of the users of the system, we have selected an operating environment designed for this purpose. We require the ability to migrate software function back and forth between the workstations and the host, depending on the locus of the data, and the amount of sharing or locking needed of the data. These requirements strongly suggest a system based around a UNIX (or UNIX-equivalent) environment. We can satisfy the requirement for running programs on either host or workstation in the same operating system environment, linked together by networks, and providing the relational database system discussed in the previous section.

Another alternative that was investigated was the use of IBM VM/CMS on a host. However the only candidate workstations available are supplied only by IBM, namely the IBM PC/370 (or AT/370) models, which can run VM/PC, a subset of the mainframe version. Unfortunately, the amount of performance one can get from a graphics workstation running this environment is quite minimal, and the wealth of graphics windowing software available for UNIX is not usable.

The conclusion is that a UNIX-based host, fully supported by the manufacturer, in native mode, is the best match for the CAPS environment.

Further details of the configuration and requirements of the host proceeds from a number of other aspects of the system. The main storage size is primarily determined so as to keep a reasonable low level of swapping and yet high response time for a program development environment of four to six simultaneous users, and with 10 or more on line graphics workstations making heavy database queries. Discussion with a number of managers of similar systems (but for other applications) indicate that 4 MB of RAM will likely ensure this behavior.

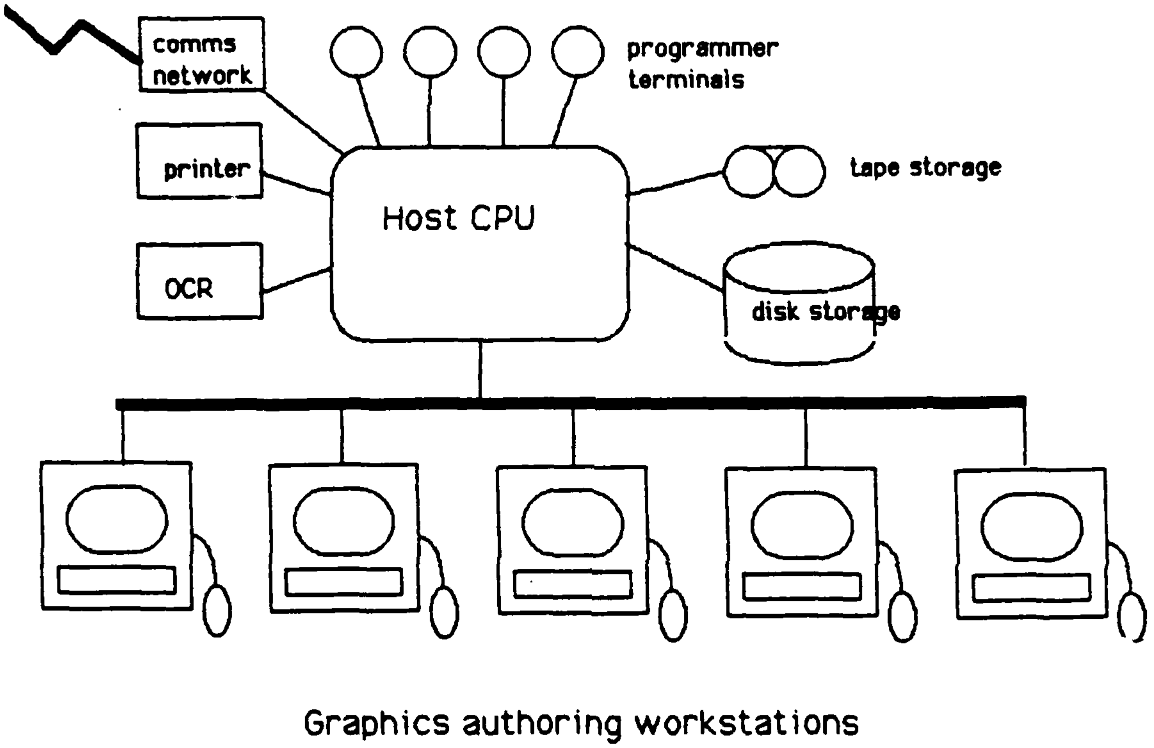


FIGURE 57. Overall CAPS System Architecture

The host must also be able to connect a number of asynchronous terminals for program editing and development. They will also be used for non-graphics related editing or file access from remote sites, in a fully deployed configuration. A controller that can run a total of up to eight terminals (or serial printers) running at 9600 or 19.2 Kbaud is needed. There are many available, such as from the manufacturer of the host CPU.

The particular choice of terminals used for this purpose is not very sensitive. One needs to have high assurance that the vendor is supplying a system that is truly VT100 compatible, and that the terminals are capable of fully working at 9600 or 19.2 Kbaud. Any such terminals can be used.

The disk and tape requirements are closely linked to the support requirements of the manufacturer of the host CPU. The minimum amount of storage required has been estimated by reviewing MTPs. They indicated a page count of about 5,000 pages, each covered with approximately 50% figures and tables and the remainder being text. Given that we will need to keep online perhaps three different versions or levels of the documents, and given internal storage inefficiencies, we estimate that between 50 to 100 megabytes of disk storage are needed for text alone.

The graphics requirement can be derived from the number of drawings needed for a prototypical MTP, and our conclusion regarding the storing of line drawings. An object oriented image description system is capable of presenting high quality images similar to the ones now in use. We are not considering going to a photographic quality of illustration in this version of the system. An average MTP would require no more than about 500 nodes to describe. If each node is made up of an 8-tuple of 4-byte parameters, each such drawing would need 16,000 bytes of storage. (Note that this imposes a lower limit on the record size being managed by the relational database manager). We estimate that about 2,500 drawings per MTP are needed; we do not see a requirement for many different generations of these drawings being online. This implies about 50 megabytes of disk storage for drawings.

Together these total in the area of 100 to 150 megabytes of storage, based on the assumption that the database manager can compact things quite well, and there are no further duplications of redundant data. When combined with the space needed for the operating system, software developed for CAPS, spooling space for print files, and other currently unspecified requirements, we conclude that 300 to 500 megabytes should be acquired.

To complement this disk storage system, we need a tape mechanism that is primarily oriented toward making backups of the online data, and more infrequently used to recover data from offline, or to move data from site to site in the future. To make practical backups, we require that the tape system be capable of backing up 500 Mbytes in about one hour. Hence the tape should write at last 10 Mbyte per minute, and each tape reel (or cartridge) must hold 100 Mbyte or more (to minimize the number of tape swaps).

Whether we require one disk drive or two, and one tape drive or two, is determined not by the operational requirements of the system, but rather by the ability of the manufacturer to maintain and support a system that has failed, and to recover data. There is no stated requirement for CAPS to be able to run 24 hours per day, in a non-stop fashion. Hence we are assuming that a vendor providing an 8 hour turnaround service policy, which can be maintained with only one disk and one tape drive meet the objectives of this program. Additional devices will add cost, and can actually be integrated into the system quite easily if it should later be concluded this is an important matter.

We are assuming that the primary output form will be hard copy masters ready for duplication as standard ARTEPs. Over time this will evolve to fully electronic forms of distribution, but initially paper is required. Hence we need a series of output devices that can handle the large number of object oriented drawings, which occupy as much as 50% of each page. This needs to be a laser printer, and realistically we need to produce 1000 pages per day at a peak, hence at least 5 ppm.

The Xerox 2700 Model II does not have the graphics capability needed to handle the task required. The Xerox 5700 does have the capability, and is also capable of higher speed use. The QMS 1200A version of the Xerox 2700 can do the needed graphics, but will run at only 12 ppm or less for complex graphics. The most recent and economic of all the laser devices is the Apple LaserWriter. This device is especially attractive because the high level graphics language that it accepts is now available as the input language to several even higher quality phototypesetters. This means that we will be able to direct the output to one of these typesetters, getting out film masters of even higher quality than a 300 lines per inch laser device can supply.

Another needed peripheral is an optical character reader, which will be used primarily for the purpose of loading the existing ARTEPs documents. This will largely be a one-time job. We anticipate converting to electronic form ONCE and not doing this multiple times. Many of the relevant background reference documents are already in floppy disk word processing form, and we will be able to bypass hard copy entirely for these documents. The prime criteria for the optical character reading system is to be able to read a reasonable variety of typewriter fonts, and not only an OCR produced font. It should support Elite, Pica, Courier, etc. The quality of the documents may not be in master quality. Remarkably high comments have been noted on the ability of the DEST reader to handle documents that are far below its officially rated quality, hence capable of inputting almost any existing typed pages.) We wish to load up the documents during a reasonable amount of time (once), and if we use the period of one week to set a scale, to load the approximate 5,000 pages we need loaded, we need a rate of 125 p/hr or more (approximately 100 char per second). It does not seem worthwhile investing in faster units than this, since the load is primarily done at the front end, and the price for faster units goes up dramatically for higher speeds. Either the DEST Turbo 203 or the Alphaword 80 Model C products appear to meet the criteria we have established.

[Work Stations. The main interface to the system will be through a high visual graphic environment. Standard ASCII text terminals are simply incapable of expressing the images and charts needed to manage the CAPS data, and to represent the quality of documents that the system is producing.

As well, to limit the speed requirements of the host processor, we have elected to use high performance workstations, each with their own processor. This processor will run its own operating system (i.e., UNIX), which will be identical to the host system, for data and task exchange, and maintenance purposes. The machines must be capable of loading data via a host network attachment at rates of 1 to 10 Mbits, since they do not have a large amount of local storage, and are requesting their data continually in a burst fashion from the CPU.

Three candidate machines have been evaluated from the graphics point of view, in terms of their monochrome resolution (that is to say, the number of resolvable points on the display screen used for data).

They are:

Apple Macintosh	512 x 342 dots
AT&T UNIX PC	720 x 340 dots
Apollo DN300	1024 x 800 dots

If we compare the total number of points on these three screens, we find that the Apple has about one fifth the number of the Apollo, and the AT&T about one third. We judge the size and detail of the Macintosh scale screen marginally low, and the UNIX and Apollo screens sufficiently detailed.

We also require that the workstations provide a standard supported pointing device, such as a mouse, and a network connection that can go to the host system. All three of these systems provide these features, but to different degrees of performance.

From the point of view of the operating environment, the Macintosh does not run UNIX in any kind of native mode. (As well, it does not have any standard operating system other than Apple's unique one as an alternative. It cannot be upgraded beyond 512K of memory, and the fastest standard output port runs at only 230 Kbits per second, far below for the other two.)

The AT&T UNIX PC has just been announced at the time of writing of this concept study. AT&T has introduced a 1 Mbit/second LAN (STARLan) for the UNIX PC. AT&T has also stated that there will be a connection to AT&T 3B machines, which are UNIX hosts normally running 3BNET, which is in fact Ethernet. The details of a direct Ethernet interface for the UNIX PCs, or alternatively the availability of a STARLan interface directly on a VAX 11/750, or equivalent gateway, need to be resolved prior to selection of this option for workstations. The Apollo-based solution is committed by the vendor at this time.

Apollo on the other hand has an extremely impressive performance network that connects its Domain workstation. It it one of the highest performance and most effective systems around, and well respected in the industry. However it is used only by Apollo, hence to connect into a host requires a gateway to translate from Apollo's network to a more industry standard choice, such as Ethernet. Given the fact that the Apollo configuration exists today, we have indicated that it is the environment of choice.

Each of the workstations should have at least 1 Mbyte of RAM memory installed, with the capability of additional amounts as a future option. This is again a major restriction on the Apple Macintosh, whereas 1 megabyte is quite standard for the other two machines mentioned.

The UNIX offerings on the three machines under consideration (DEC VAX 11/750 ULTRIX, Apollo DN300, and AT&T UNIX PC) are all derived from an AT&T licensed UNIX. Discussions with the vendors of Oracle and Ingres do not indicate any technical difficulties in supporting these configurations in a networking mode. Both point out that at this time, due to the recent announcement data of the AT&T UNIX PC, neither vendor has officially committed to support of this machine, but both are strongly leaning towards it.

ESTIMATE OF CAPS DEVELOPMENT EFFORT

In this section we describe our estimate of the R&D program that is implied by the CAPS functional requirements and system design described in the previous chapters. We describe the program stages and tasks, the estimated person-months to perform the various tasks, and the cost of acquiring the hardware for two alternative system architectures. Finally, we provide a "strawman" model of a program plan that would meet the R&D program requirements.

Program Stages and Tasks

Obviously, a number of program plans could be devised that would ultimately produce the CAPS system. The objective of the following program plan is not to constrain the R&D program to a specific program plan, but to describe the primary activities and deliverables that are essential to the overall CAPS system development.

CAPS Development Stages. A CAPS R&D program would consist of a number of program stages, during which a variety of tasks would be performed. Here we describe a program plan with six developmental stages, namely (1) front-end analysis, (2) system design, (3) hardware acquisition & software coding, (4) unit testing, (5) system integration and testing, (6) system packaging and documentation. During the first stage, the technical team members, in close coordination with the ultimate CAPS user(s), would conduct a front-end analysis to develop the detailed system objectives and functional specifications for all system modules. The output of this stage would be a detailed Functional Specifications Document, which would be a primary contract deliverable. This document would be subject to approval by the Contracting Officer's Technical Representative (COTR), and would serve as the ultimate "contract with the customer" in providing detailed technical directions to the CAPS system developers.

The second stage of system development, the system design, would focus on developing the detailed hardware and software designs. In the case of the software design, the output would typically be in the form of Programming Design Language (PDL) specifications of all program modules. The third stage, software coding would follow upon the approval, by the system development managers in consultation with the COTR, of the PDL specifications. Also during this stage, the specific hardware for CAPS would be acquired, and initial installation testing and acceptance would be performed.

The fourth stage, unit testing, would focus on evaluating and modifying as necessary each module within the CAPS system software, as well as conducting the initial steps of integrating various software modules together. This fourth step would include explicit demonstrations of individual modules, independent of other module functions that would be integrated later in the development program.

Following closely upon the fourth stage would be the fifth stage of system integration and test. This stage would probably be structured as an iterative process, with more and more system functions being integrated on a step-by-step basis. The final output of this stage would be a demonstration of all of the system functions for the entire CAPS development system. Concurrent with this integration stage, the loading of the ARTEP database would be conducted by the contractor personnel. Thus, the system demonstration at the end of this stage would include demonstrations of ARTEP materials production, based on the initial database that had been loaded.

The final stage of system development would focus on packaging the system components, both in terms of hardware and software, and on developing the system documentation. The documentation would include not only the descriptions of the system components, but also all training documents that would be used to assist the users to become fully proficient and productive with CAPS.

CAPS Development Tasks. A number of tasks, grouped into major task efforts, would be performed concurrently during the various development stages. All of these tasks are implied by the various system capabilities that have been described in previous chapters. The first major effort is concerned with developing the database management system that will be used in CAPS. Whereas a commercially-available relational DBMS (Oracle) has been specified for CAPS, several tasks must be performed to integrate this DBMS into CAPS and to provide the ARTEP data. These tasks include (1) database definition (2) database loading, and (3) inter-networking.

The second major effort in CAPS will be to design, develop, and integrate the graphics user interface. The tasks here include (4) interface development, (5) user guidance creation, (6) graphics tools development, (7) graphics creation, and (8) consistency tools creation.

The third major effort will be to develop and integrate the ARTEP materials production subsystems in CAPS. The tasks here include not only (9) authoring guides development, and (10) hard copy preparation system development, but also (11) version control creation, which will allow the CAPS system to assist the users in controlling the numerous versions to ARTEP materials that must be produced.

The fourth major effort in CAPS development will be (12) subsystems integration, in which all of the system elements are merged into the overall, integrated CAPS system.

Estimated Manpower Requirements

Table 9 shows the estimated manpower, expressed in person-months, that will be required to conduct the CAPS development tasks during the various program development stages. This estimate includes the efforts by all members of the technical development team, including program management.

TABLE 9

CAPS SYSTEM DEVELOPMENT

Estimated Manpower Requirements

DEVELOPMENT TASKS	FRONT-END ANALYSIS	SYSTEM DESIGN	HARDWARE ACQ. & CODING	UNIT TESTING	SYSTEM TESTING	PACKAGING & DOCUMENTATION	TOTALS
DBMS & NETWORK							
1. DATABASE DEFINITION	2	1	1	1	1	1	7
2. DATABASE LOADING	1	2	4	1	1	2	11
3. INTER-NETWORKING	2	2	2	1	1	1	9
USER INTERFACE							
4. USER INTERFACE DEVEL.	4	2	4	3	3	4	20
5. USER GUIDANCE	4	3	4	3	3	3	20
6. GRAPHICS TOOLS	2	2	4	3	2	3	16
7. GRAPHICS CREATION	-	1	3	1	1	3	9
8. CONSISTENCY TOOLS	2	3	2	2	2	2	13
PRODUCTION SUBSYSTEM							
9. AUTHORING GUIDES	4	3	4	3	3	3	22
10. HARD COPY PREP.	3	1	3	1	1	1	10
11. VERSION CONTROL	4	2	2	1	3	2	14
INTEGRATION							
12. SUBSYSTEM INTEGRATION	2	2	2	1	3	1	11
TOTAL	30	24	35	21	24	28	162

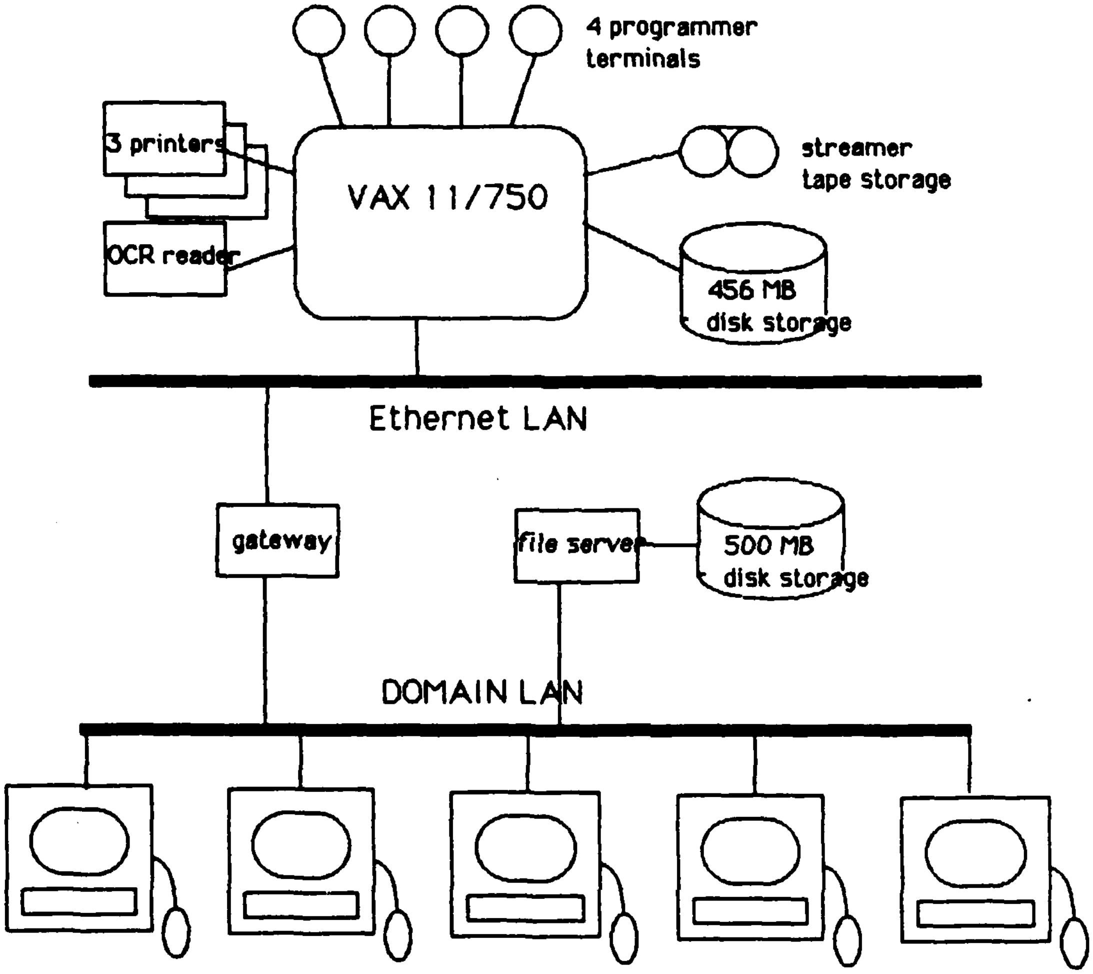
Hardware Procurement Cost Estimates

In this section we provide estimates of the procurement costs for the CAPS hardware. These estimates are based on two proposed architectures: namely, (1) a system that uses Appollo workstations, and (2) an alternative system that uses AT&T UNIX PC workstations. The technical advantages and disadvantages of these two alternatives were discussed previously.

The Apollo-based CAPS system is illustrated in Figure 58, and the associated costs are included in Table 10. The AT&T UNIX PC-based system is illustrated in Figure 59, and the associated costs for this alternative are included in Table 11.

Program Plan

Figure 60 shows the sequence of program tasks as they might be performed within a two-year system development program. Again, the tasks could be performed according to a variety of program plans. The purpose here is to illustrate the relative order in which the tasks would probably be conducted, and to indicate the general time period in which the previously estimated manpower would be expended. This program plan assumes that the manpower is evenly distributed across the various program stages.



total of 10 authoring workstations

FIGURE 58.
APPOLLO WS-BASED CAPS ARCHITECTURE

TABLE 10
COST ESTIMATE
Apollo-based CAPS Architecture

HARDWARE

Qty	Manuf/Model	Description	Price (ea $K)	Total
Computer				
1	DEC 750-XA-BE	VAX 11/750, 2 MB RAM ULTRIX incl.	60.0	60.0
1	DEC MS750-CB	2 MB additional RAM	9.0	9.0
1	DEC DZ11-DP	8 port asycn controller	2.2	2.2
1	Interlan (or equiv.)	Ethernet interface	4.0	4.0
Data Storage				
1	DEC RA81	456 MB disk storage	19.0	19.0
1	DEC TU80	1600 bpi, 100 ips tape streamer	11.0	11.0
CRT Terminals				
4	DEC VT100	or equivalent	1.9	7.6
Printers				
2	Apple LaserWriter	hard copy device	7.0	14.0
1	DEC LP32-EA (or equiv.)	600 lpm bank printer	14.0	14.0
1	DEC LA120-DA	Console printer	2.8	2.8
10	Apple ImageWriter	local printers on each workstation	0.5	5.0
Optical character reader				
1	DEST Turbo 203 or Alphaword 80 Model C	OCR reader	10.0	10.0
Graphics Terminals				
10	Apollo DN300	graphics workstations, 1 MB RAM, 800 x 1024, Domain LAN	10.0	100.0
1	Apollo DFS-500M	file server with 500 MB storage	36.0	36.0
1	Apollo COM-ETH	Ethernet gateway to VAX	3.5	3.5

SOFTWARE

Qty	Manuf/Model	Description	Price (ea $K)	Total
1	DEC QD821-HM	ULTRIX distribution	2.5	2.5
10	Apollo UNIX	UNIX license fee	0.3	3.0
1	Oracle DBMS	VAX database license	48.0	48.0
TOTAL COST				348.6

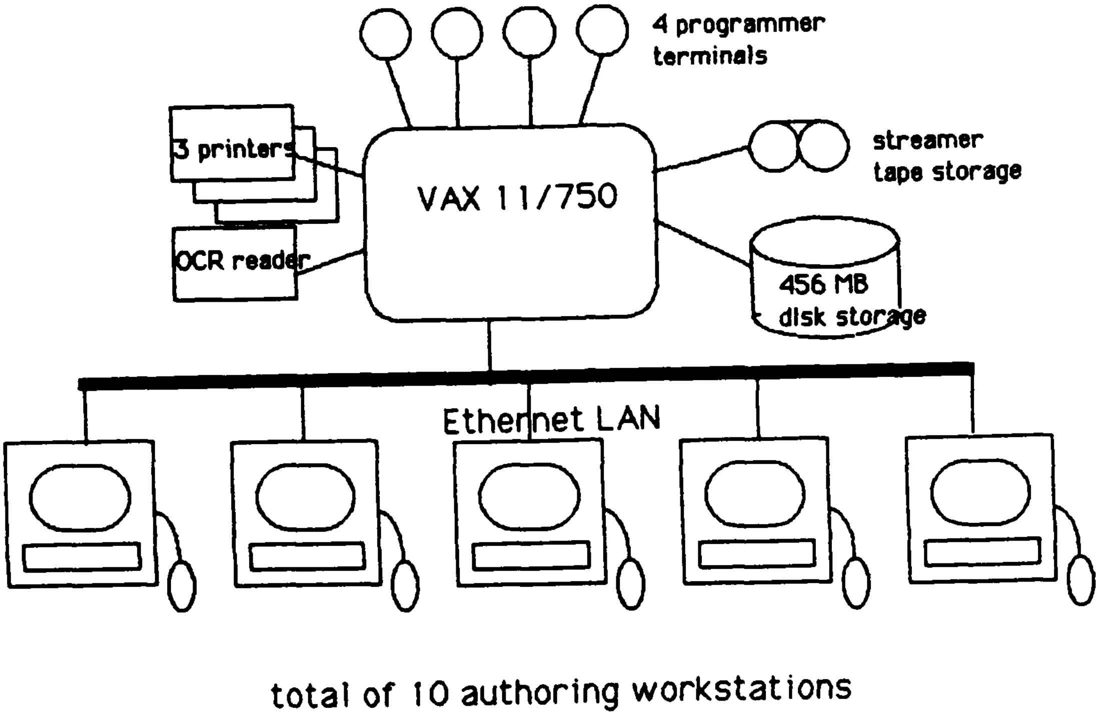


FIGURE 59.
AT&T UNIX PC-BASED CAPS ARCHITECTURE

TABLE 11
COST ESTIMATE
AT&T UNIX PC-based CAPS Architecture

HARDWARE

Qty	Manuf/Model	Description	Price (ea $K)	Total
Computer				
1	DEC 750-XA-BE	VAX 11/750, 2 MB RAM ULTRIX incl.	60.0	60.0
1	DEC MX750-CB	2 MB additional RAM	9.0	9.0
1	DEC DZ11-DP	8 port asycn controller	2.2	2.2
1	Interlan (or equiv.)	Ethernet interface	4.0	4.0
Data Storage				
1	DEC RA81	456 MB disk storage	19.0	19.0
1	DEC TU80	1600 bpi, 100 ips tape streamer	11.p0	11.0
CRT Terminals				
4	DEC VT100	or equivalent	1.9	7.6
Printers				
2	Apple LaserWriter	hard copy device	7.0	14.0
1	DEC LP32-EA (or equiv.)	600 lpm band printer	14.0	14.0
1	DEC LA120-DA	Console printer	2.8	2.8
10	Apple Imagewriter	local printers on each workstation	0.5	5.0
Optical character reader				
1	DEST Turbo 203 or Alphaword 80 Model C	OCR reader	10.0	10.0
Graphics Terminals				
10	AT&T UNIX PC	graphics workstations, 1 MB RAM, 720 x 340, 20 MB local disk	6.0	60.0
10	AT&T or ?	Ethernet interface (estimated)	1.0	10.0

SOFTWARE

Qty	Manuf/Model	Description	Price (ea $K)	Total
1	DEC QD821-HM	ULTRIX distribution	2.5	2.5
1	Oracle DBMS	VAX database license	48.0	48.0
TOTAL COST				279.1

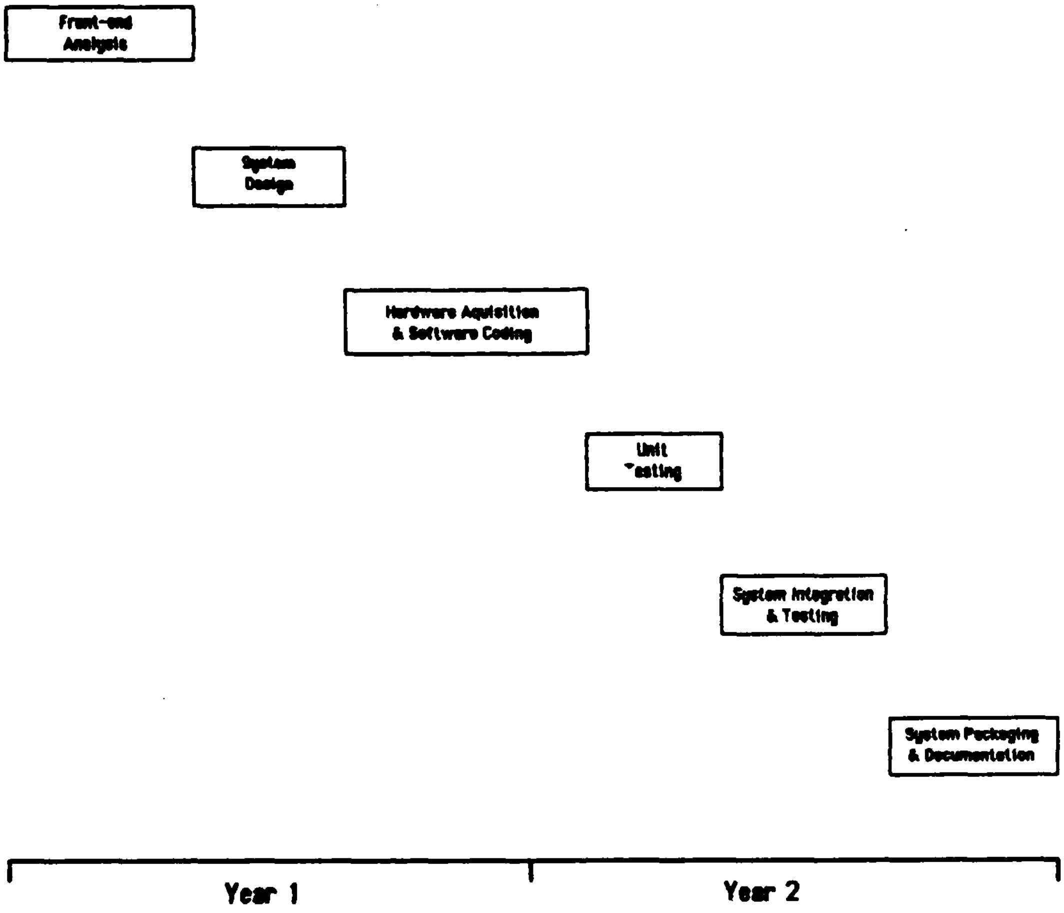


FIGURE 60.
CAPS PROGRAM PLAN